Analyse d'images pour une recherche d'images basée contenu dans le domaine transformé.

Cong Bai

▶ **To cite this version:**

Cong Bai. Analyse d'images pour une recherche d'images basée contenu dans le domaine transformé.. Autre. INSA de Rennes, 2013. Français. NNT : 2013ISAR0003 . tel-00907290

HAL Id: tel-00907290

https://theses.hal.science/tel-00907290

Submitted on 21 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Résumé

Cette thèse s'inscrit dans la recherche d'images basée sur leur contenu. La recherche opère sur des images représentées dans un domaine transformé et où sont construits directement les vecteurs de caractéristiques ou indices. Deux types de transformations sont explorés : la transformée en cosinus discrète ou Discrete Cosine Transform (DCT) et la transformé en ondelettes discrète ou Discrete Wavelet Transform (DWT), utilisés dans les normes de compression JPEG et JPEG2000. Basés sur les propriétés des coefficients de la transformation, différents vecteurs de caractéristiques sont proposés. Ces vecteurs sont mis en œuvre dans la reconnaissance de visages et de textures couleur.

Dans le domaine DCT, sont proposés quatre types de vecteurs de caractéristiques dénommés «patterns» : Zigzag-Pattern, Sum-Pattern, Texture-Pattern et Color-Pattern. Le premier type est l'amélioration d'une approche existante. Les trois derniers intègrent la capacité de compactage des coefficients DCT, sachant que certains coefficients représentent une information de directionnalité. L'histogramme de ces vecteurs est retenu comme descripteur de l'image. Pour une réduction de la dimension du descripteur lors de la construction de l'histogramme il est défini, soit une adjacence sur des patterns proches puis leur fusion, soit une sélection des patterns les plus fréquents. Ces approches sont évaluées sur des bases de données d'images de visages ou de textures couramment utilisées.

Dans le domaine DWT, deux types d'approches sont proposés. Dans le premier, un vecteur-couleur et un vecteur–texture multirésolution sont élaborés. Cette approche se classe dans le cadre d'une caractérisation séparée de la couleur et de la texture. La seconde approche se situe dans le contexte d'une caractérisation conjointe de la couleur et de la texture. Comme précédemment, l'histogramme des vecteurs est choisi comme descripteur en utilisant l'algorithme K-means pour construire l'histogramme à partir de deux méthodes. La première est le procédé classique de regroupement des vecteurs par partition. La seconde est un histogramme basé sur une représentation parcimonieuse dans laquelle la valeur des bins représente le poids total des vecteurs de base de la représentation.

Mots-clés : DCT, DWT, extraction de caractéristiques, k-means, représentation parcimonieuse, reconnaissance des visages, recherche de texture couleur.

Abstract

This thesis comes within content-based image retrieval for images by constructing feature vectors directly from transform domain. In particular, two kinds of transforms are concerned: Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), which are used in JPEG and JPEG2000 compression standards. Based on the properties of transform coefficients, various feature vectors in DCT domain and DWT domain are proposed and applied in face recognition and color texture retrieval.

The thesis proposes four kinds of feature vectors in DCT domain: Zigzag-Pattern, Sum-Pattern, Texture-Pattern and Color-Pattern. The first one is an improved method based on an existing approach. The last three ones are based on the capability of DCT coefficients for compacting energy and the fact that some coefficients hold the directional information of images. The histogram of these patterns is chosen as descriptor of images. While constructing the histogram, with the objective to reduce the dimension of the descriptor, either adjacent patterns are defined and merged or a selection of the more frequent patterns is done. These approaches are evaluated on widely used face databases and texture databases.

In the aspect of DWT domain, two kinds of approaches for color texture retrieval are proposed. In the first one, color-vector and multiresolution texture-vector are constructed, which categorize this approach into the context of extracting color and texture features separately. In contrast, the second approach is in the context of extracting color and texture features jointly: multiresolution feature vectors are extracted from luminance and chrominance components of color texture. Histogram of vectors is again chosen as descriptor and using k-means algorithm to divide feature vectors into partitions corresponding to the bins of histogram. For histogram generation, two methods are used. The first one is the classical method, in which the number of vectors that fall into the corresponding partition is counted. The second one is the proposition of a sparse representation based histogram in which a bin value represents the total weight of corresponding basis vector in the sparse representation.

Keywords: DCT, DWT, feature extraction, K-means, sparse representation, face recognition, color texture retrieval.

N° d'ordre : 13 ISAR 03 / D13-03
Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coësmes • CS 70839 • F-35708 Rennes Cedex 7
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96

Thèse

2013

Cong BAI

THESE INSA Rennes
sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du signal et de l'image

présentée par
Cong BAI
ECOLE DOCTORALE : *Matisse*
LABORATOIRE : IETR CNRS UMR 6164

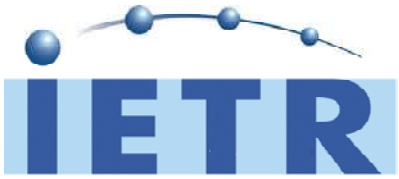
Image analysis for content based image retrieval in transform domain

Thèse soutenue le 21.02.2013
devant le jury composé de :

Philippe CARRE
Professeur des universités, Université de Poitiers / Président
Gérard POISSON
Professeur des universités, IUT de Bourges, Université d'Orléans/ Rapporteur
Nasreddine TALEB
Professeur des universités, Université Djiilali Liabes, Algérie / Rapporteur
Kidiyo Kpalma
Maître de conférences (HDR), INSA de Rennes / Co-encadrant
Joseph Ronsin
Professeur des universités, INSA de Rennes / Directeur de thèse

Image analysis for content based image retrieval in transform domain

Cong BAI



To My Parents

Thank Dr. Kidiyo KPALMA!

Thank Prof. Joseph RONSIN!

Thank China Scholarship Council!

Thank my father, Bai Weidong!

Thank my mother, Zhang Xuejun!

Thank all my friends!

Contents

1 Introduction

1.1 Background

The rapid increase in the digital image collections gives more and more information to people. At the mean time, the difficulty for an efficient use of this information is also growing, unless we can browse, search and retrieve it easily: that brings image retrieval technology. The purpose of image retrieval is to provide users with an easy access to images of interest. Image retrieval has become an active research domain since 1970s [1]. Two different angles could be found in this domain: text-based and content-based.

Text-based is the way to index images with text keywords, which was a major direction of early work on image retrieval, and comprehensive reviews are presented in [2,3]. A popular framework of text-based image retrieval is to first annotate images by manual effort: captions or embedded text, and then keyword or full text searching can be used to perform retrieval. Text-based technique is an accurate and effective method for finding annotated images. These images are organized by category, such as animals, natural scenes, people and so on. All indexing entries of image are done by human indexers who indicate the important objects in an image. Since automatically generating descriptive texts for a wide range of images is impossible, there are three major difficulties for text-based technologies: 1) huge amount of labor required in manual image annotation with "information explosion"; 2) different people may annotate differently the same image contents, this results from the rich content in the image and the subjectivity of manual annotation and may lead to unrecoverable mismatches in later retrieval processes; 3) annotations could be written in different languages.

To overcome these three difficulties, content-based image retrieval (CBIR) has been presented in the early 1990s: a new framework of image retrieval is proposed, in which

images could be indexed by their own visual contents. This research field is a diverse field, in which researchers from various disciplines, such as computer science, electrical engineering, mathematics, information sciences, physics, business, humanities, biology, medicine propose various approaches [4]. With these various researches and various approaches, CBIR can be applied in very diverse fields, including but not limited to:

- web search [5–7]
- mobile application [8,9]
- arts and museums [10,11]
- medical imaging [12,13]
- geoscience [14,15]
- business (trademark) [16–19]
- intelligent transportation [20]
- criminal prevention [21,22]

With this variety of applications, the core or the key problem of CBIR is the same: in order to find images that are visually similar to a given query, it should have both a proper representation of the images by compacting visual features and a measure that can determine how similar or dissimilar the different images are from the query. Comprehensive reviews could be found in [1,23–27]. The general diagram of CBIR is shown in Figure 1.1.

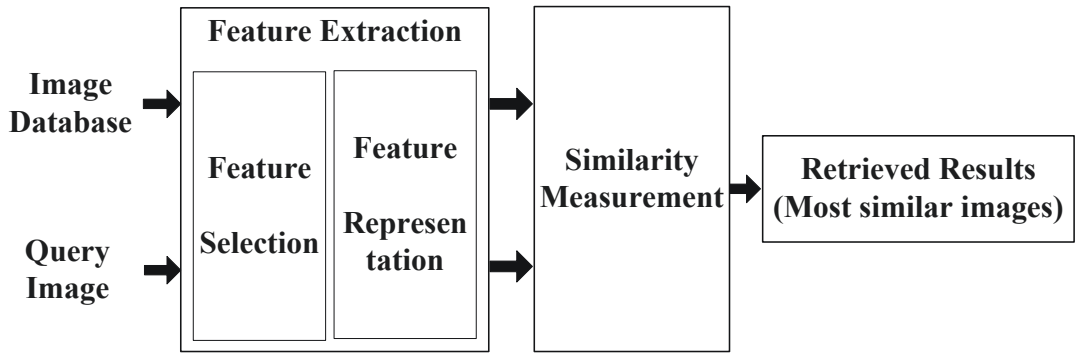


Figure 1.1: Diagram of Content-based image retrieval

Feature extraction is the basis of CBIR, which is a process of transferring the input image into the set of features (also named feature vectors). Feature vector is a reduced representation of input image. If feature vectors are chosen appropriately, they could provide enough information of the input image in order to perform the retrieval task by this reduced representation instead of the original input image. There are two main categories of feature vectors used for image retrieval [28]: intensity-based (color and texture) and

geometry-based (shape). The representation of feature vectors is called feature descriptor. A feature descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects. To obtain local visual descriptors, an image is often firstly divided into parts. Using a partition is the simplest way, which cuts the image into tiles of equal size and shape, but a simple partition does not generate perceptually meaningful regions. A better method is to divide the image into homogenous regions according to some criterion using region segmentation algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image is to perform a complete object segmentation to obtain semantically meaningful objects (like ball, car, horse). However, people tend to use high-level concepts to describe images, but features extracted by current computer vision technologies are mostly low-level features (shape, color or texture). Generally speaking, low-level features do not have a direct link to the high-level concepts. To reduce this semantic gap, some off-line and on-line processing are needed: supervised learning, unsupervised learning, a powerful and user-friendly intelligent query interface, relevance feedback technology and so on. Comprehensive reviews on reducing semantic gap can be found in [25].

Once the feature extraction is done, "how to use them for measuring the similarity between images" should be considered. The technologies can be divided into region-based similarity, global similarity, or a combination of both [26]. In each method, if features are treated as vectors, different kinds of distance in Euclidean or geodesic space are used; otherwise, if they are treated as non-vector representation, probabilistic density, for example, information divergence measures are often used.

1.2 Overview of the thesis

This thesis focuses on intensity-based image retrieval. Although semantic gap between low-level features and high-level concepts is still a key problem in CBIR, for some applications, visual similarity may in fact be more critical than semantic similarity [26]. The extraction of color and texture features is still a difficult problem and the performance of existing approaches cannot meet requirements, especially when applied on different kinds of databases. Recent reviews on color and texture retrieval can be found in [29–31].

Feature extraction is the most important step in CBIR. The scope of the thesis is

trying to find better approaches that can extract color and texture features directly from transform domains and use the combination of both to perform retrieval task. Corresponding work is meaningful as majority of images are stored in compressed format and most of compression technologies adopt different kinds of transforms to achieve compression. Traditional approaches of indexing compressed images is to decode the images to the pixel domain firstly and to adopt the approaches of image retrieval in which features are extracted from pixel domain or other kinds of transform domain to extract features instead of extracting features from transforms adopted for compression. For example, perform retrieval task on JPEG images by extracting features from wavelet transform domain, or on JPEG2000 images by constructing features descriptors from Dual-tree Complex Wavelet Transform coefficients. In our approaches, features are constructed directly in the transform domain which is the same as that is used for compression. This framework preserves advantages of reduced time consuming and low computational load.

1.3 Road map

The contents of the thesis are structured as follows: the fundamental concepts of CBIR and some theories used in our approaches are introduced in Chapter 2. Approaches in Discrete Cosine Transform domain aiming on JPEG compressed images are presented in Chapter 3 in which one improved method based on an existing approach and two new approaches are proposed. Approaches in Discrete Wavelet domains targeted for JPEG2000 compressed images are detailed in Chapter 4, in which two new approaches are proposed. Finally, conclusion and perspective is given in Chapter 5.

2

Fundamental concepts

In this chapter, the fundamental concepts concerning our approaches for CBIR in transform domain are presented:

- Applied transforms: Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT): from the coefficients of these two transforms, feature vectors are constructed.
- Feature descriptor: histogram is chosen as feature descriptor of images; data clustering (K-means algorithm) and sparse representation that are used for generating histograms
- Similarity measurements concerning measure the similarity between histograms
- Performance evaluation for CBIR.

2.1 Discrete Cosine Transform

In the last decades, Discrete Cosine Transform (DCT) has been widely adopted by video and image coding standards, for example, JPEG, MPEG and etc. Generally, images are decomposed into blocks and on which transform is performed. The transform converts spatial variations into frequency variations and the original block can be reconstructed by applying the inverse DCT. The advantage of doing this is that many of the coefficients, usually the higher frequency components can be strongly quantized and even truncated to zero. This will lead to compression efficiency in the coding stage.

In the set of coefficients belonging to a block, each coefficient is independent of each other. It is possible to obtain a reconstruction of the block of pixels while using only some parts of the coefficients. So these coefficients correspond to kind of synthesis of the contents of the block and can be used as features. Moreover, for the coefficients belonging to the kept part of coefficients used for reconstruction, their values can be simplified by

quantization. So these coefficients correspond to kind of synthesis of the contents of the block and can be used as features.

2.1.1 One-Dimensional DCT

The general definition for a one-dimensional DCT of a length N data is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos[\frac{\pi(2x+1)u}{2N}] \tag{2.1}$$

for $u = 0, 1, 2, \ldots, N-1$. And the inverse transformation is defined as:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos[\frac{\pi(2x+1)u}{2N}] \tag{2.2}$$

with $\alpha(u)$ defined as:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0 \end{cases} \tag{2.3}$$

It is clear from Equation 2.1 that for $u = 0$,

$$C(u=0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x) \tag{2.4}$$

Thus the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the DC coefficient. All other transform coefficients are called the AC coefficients.

Let's fix $N = 8$, and rewrite Equation 2.1 in the form of matrix product, we can get:

$$\mathbf{C} = \mathbf{BF} \tag{2.5}$$

where $\mathbf{C} = [C(0), C(1), \ldots, C(7)]^T$, $\mathbf{F} = [f(0), f(1), \ldots, f(7)]^T$, and

$$\mathbf{B} = \begin{pmatrix} B_0 \\ B_1 \\ \cdots \\ B_7 \end{pmatrix} = \begin{pmatrix} \alpha(0) & \alpha(0) & \cdots & \alpha(0) \\ \alpha(1)\cos[\frac{\pi}{16}] & \alpha(1)\cos[\frac{3\pi}{16}] & \cdots & \alpha(1)\cos[\frac{15\pi}{16}] \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(7)\cos[\frac{7\pi}{16}] & \alpha(7)\cos[\frac{21\pi}{16}] & \cdots & \alpha(7)\cos[\frac{135\pi}{16}] \end{pmatrix} \tag{2.6}$$

Each row of $\mathbf{B}$ can be seen as a basis function $B_u(x), x = (0, 1, \ldots, 7)$, and the figures

of these functions are shown in Figure 2.1. As we can see, the top-left waveform ($u = 0$) is simply a constant, whereas other waveforms ($u = 1, 2, \ldots, 7$) show the behavior at progressively higher frequencies [32]. These waveforms are called cosine basis function, which are orthogonal and independent [33]. In accordance with our previous description, DC coefficient $C(0)$ is the average value of $\mathbf{F}$.

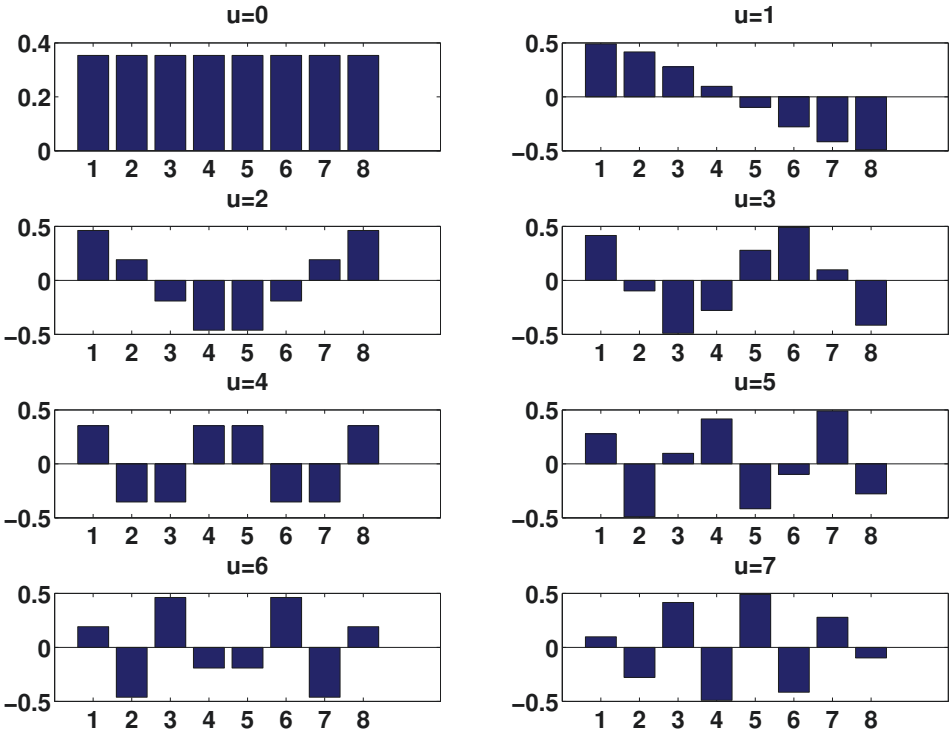


Figure 2.1: Basis functions of 1-D DCT (N=8)

2.1.2 Two-Dimensional DCT

Images are seen as two-dimensional (2-D) discrete signals, so one-dimensional DCT should be extended to a two-dimensional space when it is applied on images. The 2-D DCT for a $N \times M$ block is given by:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \qquad (2.7)$$

for $u = 0, 1, 2, \ldots, N-1,\ v = 0, 1, 2, \ldots, M-1$, and the inverse transform is defined as:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \alpha(u)\alpha(v)C(u,v) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \tag{2.8}$$

for $x = 0, 1, 2, \ldots, N-1,\ y = 0, 1, 2, \ldots, M-1$, and $\alpha(o), o \in \{u, v\}$ is defined as:

$$\alpha(o) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad o = 0 \\ \sqrt{\frac{2}{N}} & for \quad o \neq 0 \end{cases} \tag{2.9}$$

In Equation 2.7,

$$B(u,v) = \alpha(u)\alpha(v) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \tag{2.10}$$

are the basis functions of 2-D DCT. These basis functions can be generated by multiplying the horizontally oriented set of cosine basis functions with vertically oriented set of cosine basis functions:

$$B(u,v) = B_u B_v^T \tag{2.11}$$

With these basis functions, the inverse transform of DCT (Equation 2.8) can be rewritten as:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} C(u,v)B(u,v) \tag{2.12}$$

From this equation, explanations on why some coefficients could be used to construct descriptor of the image can be found: we can see that image $f(x,y)$ can be seen as a weighted sum of basis functions $B(u,v)$, with the $C(u,v)$ as weights. That's to say each coefficient gives the weight of a kind of structural patterns corresponding to a basis function used on reconstructing the contents of the pixels blocks. This opens the interest of DCT coefficients for image retrieval.

The basis functions for $N = M = 8$ are shown in Figure 2.2. For purpose of illustration, it is plotted as a grey scale image: the smaller the value is, the darker it is plotted; the larger the value is, the lighter it is plotted. It can be noted that the basis functions of 2-D DCT exhibit a progressive increase in frequency both in the vertical and horizontal direction: horizontal frequencies increase from left to right, and vertical frequencies increase from top to bottom. Similar with 1-D DCT, DC coefficient at the upper left is the average value of the block and the AC coefficients contain the progressive increase frequency information both in the vertical and horizontal direction. From aforementioned analysis, coefficients

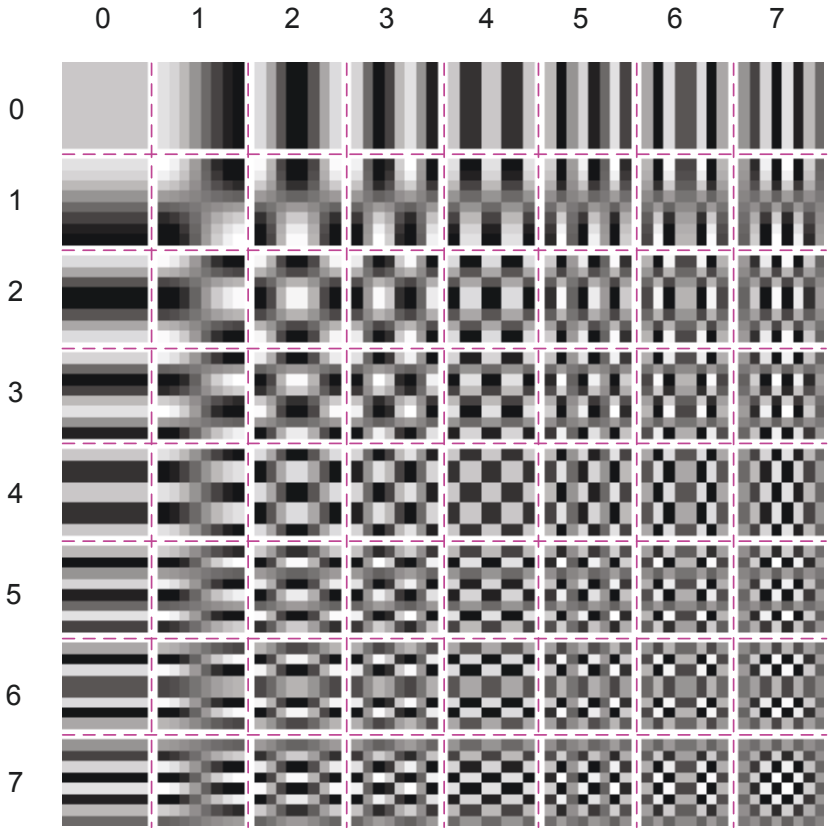
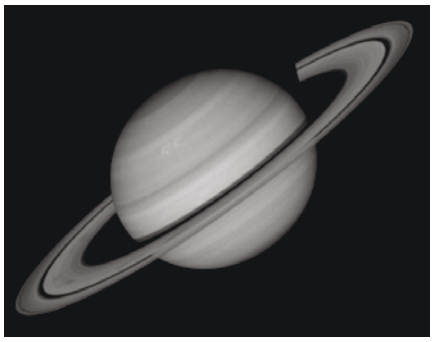


Figure 2.2: Basis functions of two dimensional DCT (N=M=8)

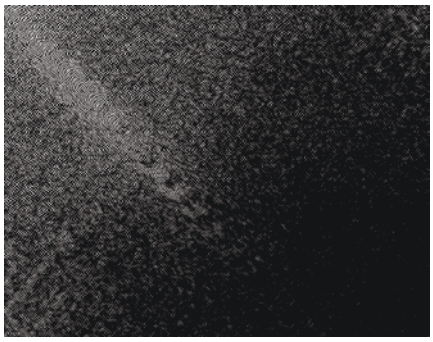
$C(0,v), v = (1,2,\ldots,7)$ will reflect the horizontal information of the image, coefficients $C(u,0), u = (1,2,\ldots,7)$ will reflect the vertical information and coefficients $C(u,v), u = v = (1,2,\ldots,7)$ will reflect the diagonal information.

In Figure 2.3(a), the original image of "Saturn" is shown, and in Figure 2.3(b), the coefficients of DCT applied on the whole images are plotted. As the "Saturn" image has "stronger" information in diagonal direction than in horizontal and vertical directions, there is strong energies in diagonal direction than in horizontal and vertical directions. Figure 2.3(c) and Figure 2.3(d) give the comparison of coefficients between 4×4 and 8×8 block DCT applied on image.

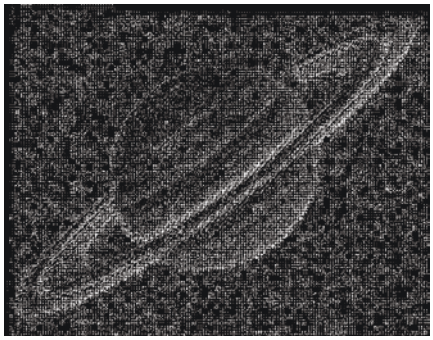
The 4×4 block DCT transform are not used in JPEG compression standard because their efficiency is lower than the one with 8×8 block, but it can be observed that the coefficients of 4×4 block DCT give more perceptual information than that of 8×8 block DCT. And our objective is to extract descriptors of the image contents in the compressed domain, so we finally choose the coefficients of 4×4 block DCT to construct the feature vectors in the approaches aimed on DCT transformed images. We will show that



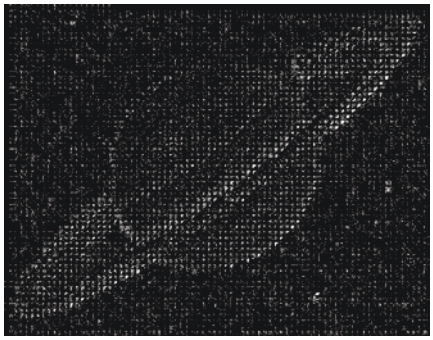
(a) Saturn



(b) DCT of Saturn



(c) 4×4 block DCT of Saturn



(d) 8×8 block DCT of Saturn

Figure 2.3: DCT of Saturn

coefficients of 4×4 block DCT can be extracted directly from coefficients of 8×8 DCT block.

2.1.3 From 8×8 DCT to 4×4 DCT

As our proposals are based on the coefficients of 4×4 block DCT, but 8×8 block is adopted in JPEG compression standard, we will propose the way of extracting 4×4 block DCT coefficients directly from 8×8 block DCT coefficients in this section, inspired from the proposal for converting 8×8 DCT to 4×4 Integer-transform [34].

The problem is defined as having an 8×8 DCT block D transformed from image block d, it could be converted to four 4×4 blocks D_{ij} that satisfies:

$$D_{ij} = T_4 d_{ij} T_4^T \tag{2.13}$$

where T_4 is 4-point DCT matrix, d_{ij} are four sub-blocks of d in size of 4×4 in the order

of left to right and up to bottom, $i, j = \{1, 2\}$.

As every 4×4 image block can be derived from 8×8 image block through a pair of matrix multiplication:

$$d_{ij} = e_i d e_j^T \tag{2.14}$$

where e_1 and e_2 are 4×8 matrices that are defined as the upper and lower half of an 8×8 identity matrix respectively.

And the image block d can be reconstructed using 8×8 inverse DCT:

$$d = T_8^T D T_8 \tag{2.15}$$

where T_8 is 8-point 1-D DCT matrix.

So substituting Equation 2.15 into Equation 2.14 and then into Equation 2.13, we could get:

$$D_{ij} = T_4 e_i T_8^T D T_8 e_j^T T_4^T \tag{2.16}$$

let

$$E_i = T_4 e_i T_8^T \tag{2.17}$$

we have

$$D_{ij} = E_i D E_j^T \tag{2.18}$$

According to definition of 1-D DCT described in Equation 2.1, the N-point 1-D DCT matrix T_N is defined as:

$$\begin{aligned} t_{1j} &= \sqrt{\frac{1}{N}} && j = (1, 2, \ldots, N) \\ t_{ij} &= \sqrt{\frac{2}{N}} \cos\left(\frac{(2j-1)(i-1)}{2N}\right) && i = (2, \ldots, N), j = (1, 2, \ldots, N) \end{aligned} \tag{2.19}$$

They could be written in the matrix form:

$$T_4 = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 \end{pmatrix} \tag{2.20}$$

$$T_8 = \begin{pmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{pmatrix} \tag{2.21}$$

So

$$E_1 = \begin{pmatrix} 0.7071 & 0.6407 & 0 & -0.2250 & 0 & 0.1503 & 0 & -0.1274 \\ 0.4531 & 0.5 & 0.2079 & 0 & -0.0373 & 0 & 0.0114 & 0 \\ 0 & 0.2079 & 0.5 & 0.3955 & 0 & -0.1762 & 0 & 0.1389 \\ -0.1591 & 0 & 0.3955 & 0.5 & 0.2566 & 0 & -0.0488 & 0 \end{pmatrix} \tag{2.22}$$

$$E_2 = \begin{pmatrix} 0.7071 & -0.6407 & 0 & 0.2250 & 0 & -0.1503 & 0 & 0.1274 \\ 0.4531 & -0.3266 & -0.2079 & 0.3266 & -0.0373 & -0.1353 & -0.0114 & 0.1353 \\ 0 & 0.2079 & -0.5000 & 0.3955 & 0 & -0.1762 & 0 & 0.1389 \\ -0.1591 & 0.3266 & -0.3955 & 0.1353 & 0.2566 & -0.3266 & 0.0488 & 0.1353 \end{pmatrix} \tag{2.23}$$

As the number of non-zero entries in E_1 and E_2 are smaller than that of in T_8 and T_8^T, the complexity for calculating Equation 2.18 is lower than calculating Equation 2.15 and this complexity could be further reduced by factorization of DCT transformation matrix [34]. From this point of view, our proposals have lower complexities than the methods that retrieve JPEG images in pixel domain or the methods that retrieve JPEG images in wavelet domain or other transform domains: for executing those methods, they should use Equation 2.15 to decompress images firstly to get the datas for constructing feature vector and for executing our proposals, we can get the datas for constructing feature vectors by Equation 2.18.

In next section, the transform used in JPEG2000 will be introduced and analyzed in the aspect of feature representation.

2.2 Discrete Wavelet Transform

Wavelets in a certain sense ideally embody the idea of locality by resorting to localized bases, which are organized according to different scales or resolutions [35] [36]. The wavelet coefficients of a generic piecewise smooth image are mostly negligible except for those along

important visual cues such as jumps or edges. Thus wavelets are efficient tools for adaptive image analysis and data compression [37].

Wavelet transform decomposes images into component waves of varying spatial extensions, called wavelets. These wavelets are localized variations of detail in an image. They can be used for a wide variety of fundamental signal processing tasks, such as compression, removing noise, or extracting features from images. The heart of wavelet analysis is multiresolution analysis that is the decomposition of a image into subimages of different resolution levels. [38].

With different wavelets, different kinds of wavelet transforms can be defined. In this section, we start from Haar wavelet, which is the simplest type of wavelet and it is related to Haar transform which serves as a prototype for all other wavelet transforms. Moreover Haar transform coefficients expresses direct link between coefficients and pixels. It uses simple coefficients for performing local analysis. Knowing well the Haar transform in details will make it easy to understand more complicated wavelet transforms.

Here we address the wavelet transform with discrete signals, which is defined as follows:

$$\mathbf{f} = (f_1, f_2, \ldots, f_N) \tag{2.24}$$

where N is a positive integer referred as the length of $\mathbf{f}$.

The scalar product of two discrete signals $\mathbf{f} = (f_1, f_2, \ldots, f_N)$ and $\mathbf{g} = (g_1, g_2, \ldots, g_N)$ is defined as:

$$\mathbf{f} \cdot \mathbf{g} = f_1 g_1 + f_2 g_2 + \cdots + f_N g_N \tag{2.25}$$

2.2.1 Haar wavelets

The Haar transform can decompose a discrete signal into two subsignals of half length. One subsignal is a running average or trend and the other one is a running difference or fluctuation.

The first trend $\mathbf{a}^1 = (a_1, a_2, \cdots, a_{N/2})$, for the signal $\mathbf{f}$ is computed by taking a running average in the following way.

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \tag{2.26}$$

for $m = 1, 2, 3, \cdots, N/2$.

The first fluctuation $\mathbf{d}^1 = (d_1, d_2, \cdots, d_{N/2})$ is defined as:

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \tag{2.27}$$

for $m = 1, 2, 3, \cdots, N/2$.

The Haar wavelet transform can be performed on several levels, with the previous definitions first trend and first fluctuation, the first level of Haar wavelet transform is defined as a mapping $\mathbf{H}_1$:

$$\mathbf{f} \xrightarrow{\mathbf{H}_1} (\mathbf{a^1} \mid \mathbf{d^1}) \tag{2.28}$$

which maps a discrete signal $\mathbf{f}$ to its first trend $\mathbf{a^1}$ and its first fluctuation $\mathbf{d^1}$.

If the 1-level Haar wavelets are defined as

$$\begin{array}{rcccccccccc}
\mathbf{W}_1^1 & = & (\frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & \dots & \dots & \dots & \dots & 0) \\
\mathbf{W}_2^1 & = & (0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & \dots & \dots & \dots & 0) \\
\mathbf{W}_3^1 & = & (0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & \dots & 0) \\
\vdots & & & & & & & & & & \\
\mathbf{W}_{N/2}^1 & = & (0 & 0 & \dots & \dots & \dots & \dots & \dots & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}})
\end{array} \tag{2.29}$$

and 1-level Haar scaling signals are defined as

$$\begin{array}{rcccccccccc}
\mathbf{V}_1^1 & = & (\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \dots & \dots & \dots & \dots & 0) \\
\mathbf{V}_2^1 & = & (0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \dots & \dots & \dots & 0) \\
\mathbf{V}_3^1 & = & (0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \dots & 0) \\
\vdots & & & & & & & & & & \\
\mathbf{V}_{N/2}^1 & = & (0 & 0 & \dots & \dots & \dots & \dots & \dots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}})
\end{array} \tag{2.30}$$

Then the fluctuation subsignals $\mathbf{d^1} = (d_1, d_2, \cdots, d_{N/2})$ could be defined as scalar products with the 1-level Haar wavelets:

$$d_m = \mathbf{f} \cdot \mathbf{W}_m^1 \tag{2.31}$$

for $m = 1, 2, \ldots, N/2$.

And the first trend subsignals $\mathbf{a^1} = (a_1, a_2, \cdots, a_{N/2})$ could be expressed as scalar

products with 1-level Haar scaling signals:

$$a_m = \mathbf{f} \cdot \mathbf{V}_m^1 \tag{2.32}$$

for $m = 1, 2, \ldots, N/2$.

The ideas discussed above could extend to every level. The 2-level scaling and wavelets are defined as follows:

$$\begin{array}{rcl}
\mathbf{V}_1^1 & = & (\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad 0 \quad 0 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad 0) \\
\mathbf{V}_2^1 & = & (0 \quad 0 \quad 0 \quad 0 \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad 0 \quad \ldots \quad 0) \\
\vdots & & \\
\mathbf{V}_{N/4}^1 & = & (0 \quad 0 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}})
\end{array} \tag{2.33}$$

$$\begin{array}{rcl}
\mathbf{W}_1^1 & = & (\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad 0 \quad 0 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad 0) \\
\mathbf{W}_2^1 & = & (0 \quad 0 \quad 0 \quad 0 \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad 0 \quad \ldots \quad 0) \\
\vdots & & \\
\mathbf{W}_{N/4}^1 & = & (0 \quad 0 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}})
\end{array} \tag{2.34}$$

So the 2-level trend and fluctuation are defined as follows:

$$\mathbf{a^2} = (\mathbf{f} \cdot \mathbf{V}_1^2, \mathbf{f} \cdot \mathbf{V}_2^2, \ldots, \mathbf{f} \cdot \mathbf{V}_{N/4}^2) \tag{2.35}$$

$$\mathbf{d^2} = (\mathbf{f} \cdot \mathbf{W}_1^2, \mathbf{f} \cdot \mathbf{W}_2^2, \ldots, \mathbf{f} \cdot \mathbf{W}_{N/4}^2) \tag{2.36}$$

and so on for following levels.

With different definitions of scaling signals and wavelets, several wavelet transform can be got, for example, Cohen-Daubechies-Feauveau 9/7 (CDF 9/7) wavelet [39].

2.2.2 CDF 9/7 Wavelets

CDF 9/7 is an important biorthogonal wavelet transform defined in the same way as the Haar wavelet transform by computing running averages and difference via scalar products with scaling signals and wavelets. It is called biorthogonal for two reasons: (1) the CDF 9/7 wavelet transform is not energy preserving (2) one set of basis signals is used for calculating transform values while a second set of basis signals is used for multiresolution

analysis of a signal, and these two bases are related by biorthogonality conditions.

The scaling coefficients and wavelet coefficients used in CDF 9/7 wavelet transforms are defined as:

$$\begin{aligned}
\alpha_1 &= 0.0378284554956993 & \beta_1 &= 0.0645388826835489 \\
\alpha_2 &= -0.0238494650131592 & \beta_2 &= -0.0406894176455255 \\
\alpha_3 &= -0.110624404085811 & \beta_3 &= -0.418092272881996 \\
\alpha_4 &= 0.377402855512633 & \beta_4 &= 0.788485616984644 \\
\alpha_5 &= 0.852698678836979 & \beta_5 &= -0.418092272881996 \\
\alpha_6 &= 0.377402855512633 & \beta_6 &= -0.0406894176455255 \\
\alpha_7 &= -0.110624404085811 & \beta_7 &= 0.0645388826835489 \\
\alpha_8 &= -0.0238494650131592 & & \\
\alpha_9 &= 0.0378284554956993 & &
\end{aligned} \tag{2.37}$$

Using these numbers, the typical scaling signals are:

$$\mathbf{V}_k^1 = (0, \ldots, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, 0, \ldots, 0) \tag{2.38}$$

with $\mathbf{V}_{k+1}^1$ a translation by two time-units of $\mathbf{V}_k^1$.

Likewise, the typical analysis wavelet is

$$\mathbf{W}_k^1 = (0, \ldots, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, 0, \ldots, 0) \tag{2.39}$$

with $\mathbf{W}_{k+1}^1$ a translation by two time-units of $\mathbf{W}_k^1$.

A major strength of the CDF 9/7 system is that whenever the values of a signal are closely approximated by either a constant sequence, a linear sequence, a quadratic sequence , or a cubic sequence, over the support of a CDF 9/7 analyzing wavelet, then the fluctuation value produced by the scalar product of that wavelet with the signal will closely approximate 0. Furthermore, the trend values at any given level are often close matches of an analog signal, this property makes the interpretation of trend values easily, especially in image processing [40].

2.2.3 Two-dimensional wavelets

Until now we have talked about wavelet on one-dimensional signals, so we will provide a basic summary of two-dimensional wavelet transform for image application.

A discrete image $\mathbf{f}$ is an array of M rows and N columns of real numbers:

$$\mathbf{f} = \begin{pmatrix} f_{1,1} & f_{2,1} & \cdots & f_{N,1} \\ f_{1,2} & f_{2,2} & \cdots & f_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1,M} & f_{2,M} & \cdots & f_{N,M} \end{pmatrix} \tag{2.40}$$

It is often helpful to view an image in one of two other ways. First, as a single column consisting of M signals having length N

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_M \end{pmatrix} \tag{2.41}$$

with the rows being the signals:

$$\mathbf{f}_m = (f_{1,m}, f_{2,m}, \ldots, f_{N,m}) \tag{2.42}$$

where $m = 1, \ldots, M$.

Second, as a single row consisting of N signals of length M, written as columns,

$$\mathbf{f} = (\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^N) \tag{2.43}$$

with the columns being the signals

$$\mathbf{f}_n = \begin{pmatrix} f_{n,1} \\ f_{n,2} \\ \vdots \\ f_{n,M} \end{pmatrix} \tag{2.44}$$

$n = 1, \ldots, N$.

A 2-D wavelet transform of an image $\mathbf{f}$ on 1-level can be defined, using any of the 1-D

wavelet transforms, by performing the following two steps.

Step 1. Perform a 1-level, 1-D wavelet transform, on each row of $\mathbf{f}$, thereby producing a new image.

Step 2. On the image obtained from step 1, perform the same 1-D wavelet transform on each of its columns.

It is not difficult to show that Steps 1 and 2 could be done in reverse order and the result would be the same. A 1-level wavelet transform of an image $\mathbf{f}$ can be symbolized as follow:

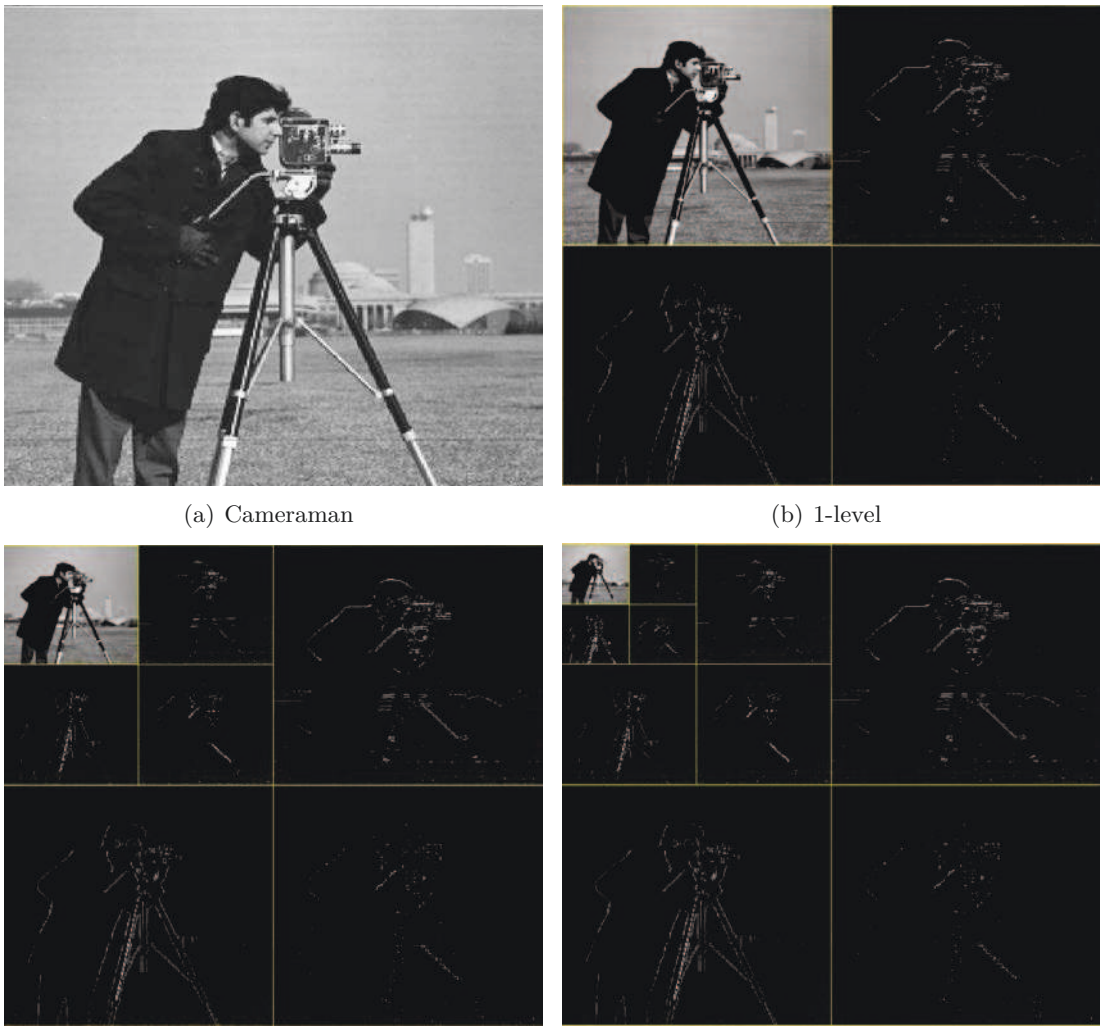
$$\mathbf{f} \mapsto \begin{pmatrix} \mathbf{a}^1 & \mathbf{h}^1 \\ \mathbf{v}^1 & \mathbf{d}^1 \end{pmatrix} \tag{2.45}$$

where the subimages $\mathbf{h}^1$, $\mathbf{d}^1$, $\mathbf{a}^1$ and $\mathbf{v}^1$ have M/2 rows and N/2 columns.

The subimage $\mathbf{a}^1$ is created by computing trends along rows of $\mathbf{f}$ followed by computing trends along columns; so it is an average, lower resolution version of the image $\mathbf{f}$. The $\mathbf{h}^1$ subimages is created by computing trends along rows of the image $\mathbf{f}$ followed by computing fluctuations along columns. Consequently, wherever there are horizontal edges in an image, the fluctuations along columns are able to detect these edges. The subimage $\mathbf{v}^1$ is similar to $\mathbf{h}^1$, except that the roles of horizontal and vertical are reversed. The subimage $\mathbf{d}^1$ tends to emphasize diagonal features, because it is created from fluctuations along both rows and columns. For example, in Figure 2.4(a), we show the original image "cameraman", and in Figure 2.4(b) its 1-level CDF 9/7 transform. The $\mathbf{a}^1$ subimage appears in the higher left quadrant of CDF 9/7 transform and it is clearly a lower resolution version of the original cameraman image. The $\mathbf{h}^1$ subimage appears in the higher right quadrant in which the horizontal edges can be seen clearly. The subimage $\mathbf{v}^1$ is shown in the lower left quadrant in which horizontal edges of cameraman is suppressed and vertical edges are emphasized. The subimage $\mathbf{d}^1$ appears in the lower right quadrant of the image in which the diagonal details are emphasized.

It should be noted that the basic principles discussed previously for 1-D wavelet analysis still apply here in the 2-D setting. For example, the fact that fluctuation values are generally much smaller than trend values is still true. In fact, in order to make the values for $\mathbf{h}^1$, $\mathbf{d}^1$ and $\mathbf{v}^1$ visible, they are displayed on a logarithmic intensity scale, while the values for the trend subimages $\mathbf{a}^1$ are displayed using an ordinary, linear scale.

As in 1-D, a K-level transform is defined by performing a 1-level transform on the previous trend $\mathbf{a}^{K-1}$ while the fluctuations subimages of all levels $\mathbf{h}^k$, $\mathbf{d}^k$ and $\mathbf{v}^k$, $k =$



(a) Cameraman (b) 1-level

(c) 2-level (d) 3-level

Figure 2.4: 2-D wavelet transform of cameraman

$(1, 2, \ldots, K)$ remain unchanged. For example, a 2-level wavelet transform is executed by computing a 1-level transform of the trend subimage $\mathbf{a}^1$ as follows:

$$\mathbf{a}^1 \mapsto \begin{pmatrix} \mathbf{a}^2 & \mathbf{h}^2 \\ \mathbf{v}^2 & \mathbf{d}^2 \end{pmatrix} \tag{2.46}$$

where the subimages $\mathbf{h}^2$, $\mathbf{d}^2$, $\mathbf{a}^2$ and $\mathbf{v}^2$ have M/4 rows and N/4 columns.

In image processing, K-level wavelet transform is often expressed in the form as shown in Figure 2.5 (in the example of 2-levels transform). In Figure 2.4(c), 2-level CDF 9/7 wavelet transform of "cameraman" is shown and Figure 2.4(d) shows its 3-level CDF 9/7 wavelet transform.

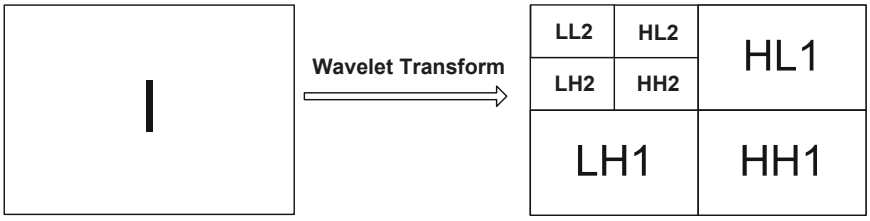


Figure 2.5: 2-levels of wavelet transform of an image

2.3 Histogram

Histogram is chosen as the feature descriptor of images in our proposals, so in this section, some fundamentals will be introduced.

From the point of view of statistics, a histogram is a function $\{h_i\}$ that counts the number of occurrence of a given element in a given set represented by bins. Thus, if we let k be the total number of occurrence of elements and N be the total number of bins, the histogram $\{h_i\}$ meets the following property:

$$k = \sum_{i=1}^{N} h_i \tag{2.47}$$

From the point of view of feature representation, a histogram $\{h_i\}$ is a mapping between a set of d-dimensional vectors and a set of non-negative real values. By this mapping, these vectors are typically represented by bins, indexed by i, which corresponds to fixed partitions of vectors. The associated reals are a measure of the mass of the vectors that fall into the corresponding partitions. For instance, in a grey-level histogram, $d = 1$, the set of possible grey values is split into N intervals, and h_i is the number of pixels in an image that have a grey value in the interval indexed by i.

Features like color and texture usually vary significantly because of inherent object variations and different illumination. Therefore it is reasonable to describe an image by a feature distribution instead of individual feature vectors. Histograms are used for approximating probability distributions, which have been successfully used for CBIR in the past. Two different categories can be found for modeling probability densities [41]:

- Parametric methods, in which a pre-determined statistical model is assumed, and the model itself contains several parameters which are optimized by fitting the model to histogram.
- Non-parametric methods, in which no specific model is pre-determined and the form

of density is determined entirely by the data. The drawback of these methods is that the representation of the model, histogram bins, for instance, could become very large.

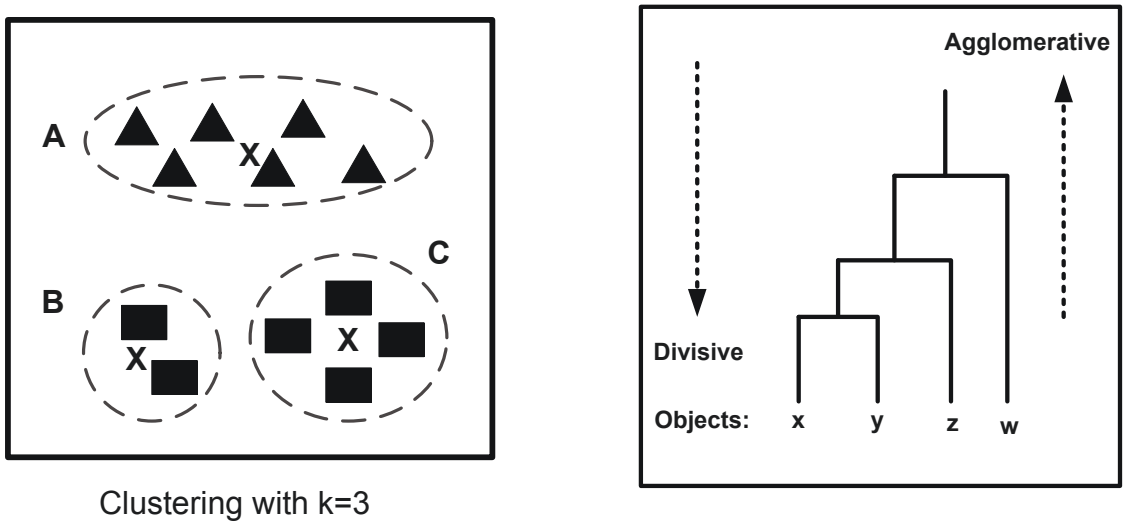
Generally speaking, we do not have a priori knowledge about the image content in image retrieval. So, we choose non-parametric histogram based method for image retrieval in this thesis. The main advantage of histogram is that it can be generated easily and quickly. Furthermore, as a global descriptor of the image, it could be insensitive to rotation of objects in images. In contrast, the main disadvantage of histogram is that it could be a bad representation if bins or the number of bins are chosen inappropriately.

In following two sections, two theories used for generating histograms will be presented: data clustering that could be used to find appropriate partitions of vectors and sparse representation that could be used to calculate the values of bins.

2.4 Data clustering

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image processing and bio-informatics. The goal of data clustering, also known as cluster analysis, is to discover the natural groups of a set of objects. These objects could be numbers, vectors or patterns and many others. More precisely, data clustering groups objects into clusters such that the similarities between objects of the same group are high while the similarities between objects of different groups are low.

Clustering algorithms can be broadly divided into two groups: hierarchical and partitional [42]. Hierarchical algorithms find successive clusters using previously established clusters, which can be either in agglomerative mode (beginning with each data point as a separate cluster and merging the most similar pair of clusters successively to form a cluster hierarchy) or in top-down mode (beginning with all the data points in one cluster and recursively dividing it into smaller clusters). Different from hierarchical clustering algorithms, partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. Figure 2.6 gives examples of these two groups (duplicate from [43]). Figure 2.6(a) gives an example of a hierarchical algorithms performed in same data sets, in which "X" indicates the center of clusters. Figure 2.6(b) shows the procedure of agglomerative clustering algorithms.



(a) Partitional (b) Hierarchical

Figure 2.6: Examples of the classic clustering algorithms

The most popular and the simplest partitional algorithm is K-means. Even though K-means has been proposed for over 50 years, it is still one of the most widely used algorithms for clustering, especially in pattern recognition and image processing. The main reasons for its popularity may due to its easy implementation, simplicity and efficiency. We will introduce this algorithm which is also adopted in our proposals for color texture retrieval in wavelet domain.

2.4.1 K-means algorithm

Let $\mathbf{X} = \{\mathbf{x_i}\}$, $i = 1, 2, \ldots, N$ be the set of N d-dimensional vectors to be clustered into a set of K clusters $\mathbf{C} = \{\mathbf{c_k}\}, k = 1, 2, \ldots, K$. ($K$ must be decided in priority.) The goal of K-means algorithm is to find a partition such that the squared error between the center of each cluster and the objects in the cluster is minimized:

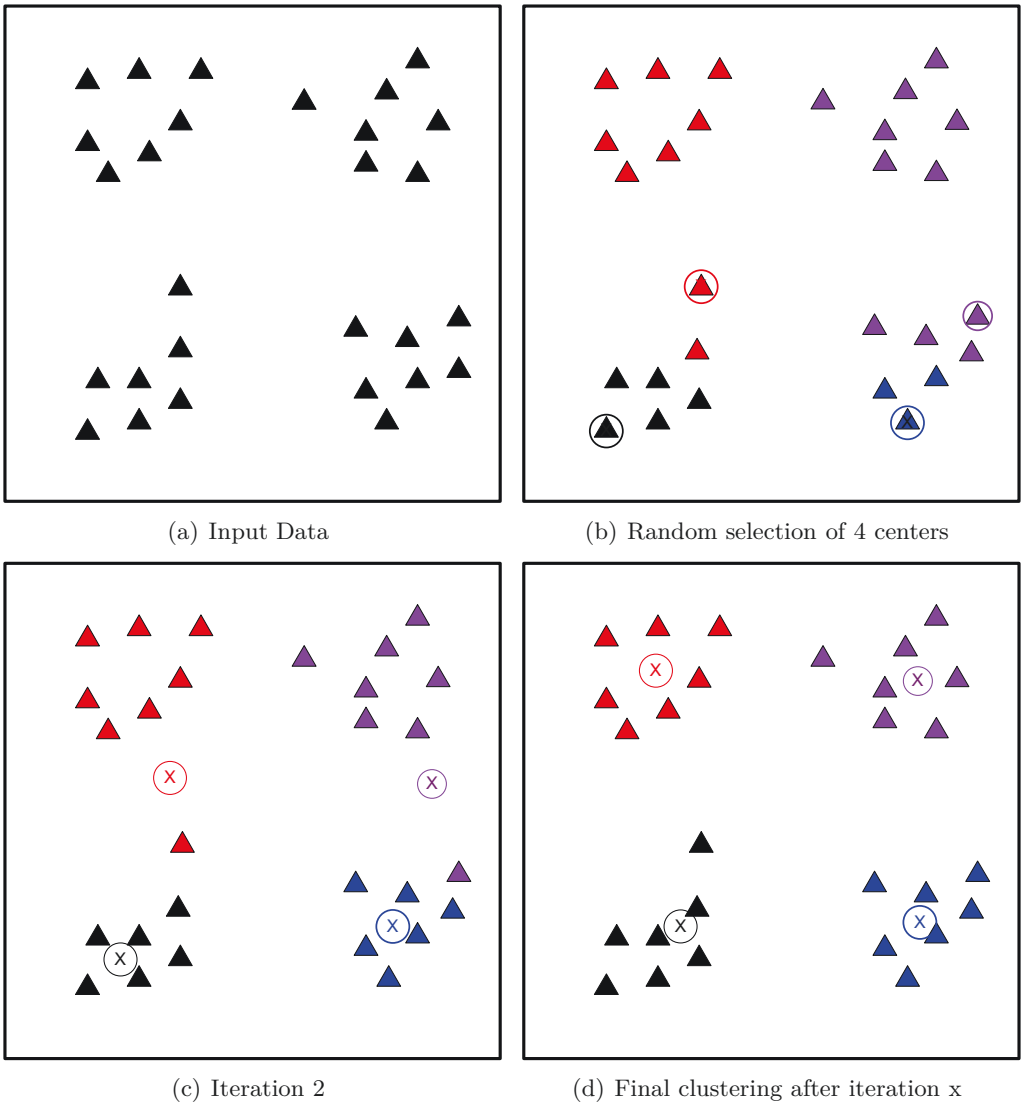
$$E = \sum_{k=1}^{K} \sum_{\mathbf{x_i} \in \mathbf{c_k}} |\mathbf{x_i} - \mathbf{D}_k|^2 \tag{2.48}$$

where $\mathbf{D}_k$ is the center of cluster $\mathbf{c_k}$. Minimizing this objective function is an NP-hard problem [44]. Thus K-means can only converge to a local minimum. K-means starts with a random partition with K centers of clusters and assign objects to clusters so as to reduce the squared error. Since the squared error always decreases with an increase in the number of clusters K (with $E = 0$ when $K = N$), it can be minimized only for a fixed number of

clusters. The main steps of K-means algorithm are as follows:

1. Assume a random partition with K centers.
2. Assign each object to the closest center.
3. Update the center of each cluster.
4. Repeat steps 2 and 3 until stability: no object move between groups.

Figure 2.7 shows a demonstration of K-means algorithm on a data set with four clusters.



(a) Input Data (b) Random selection of 4 centers

(c) Iteration 2 (d) Final clustering after iteration x

Figure 2.7: Demonstration of K-means algorithm

The K-means algorithm requires three initiated parameters: number of clusters K, cluster initialization, and distance metric. The most critical and the most difficult problem is to determine K. Although no perfect mathematical criterion exists, numerous approaches

for determining K by objective criterions have been presented [45–47]. Besides these objective criterions, "how to decide which value of K leads to the most meaningful clusters" is also an unsolved problem. However, the choice of the appropriate approach is out of the scope of this thesis: the simple choice is to find the best K empirically.

In the aspect of distance metric, K-means typically uses the Euclidean space for computing the distance between objects and cluster centers, which brings on spherical or ball-shaped clusters in data set. In this thesis, L_2 distance is chosen for measuring the similarity between objects and cluster centers.

As K-means only converges to local minima, different initializations can lead to different final clustering, and K-means is used to generate the partitions of the feature vectors and these partitions are used to generate the histogram used as feature descriptor for image retrieval in our approaches in wavelet domain, so different partitions will lead to different histograms, and finally lead to different performance when image retrieval is performed. For objective comparison between our approaches with other methods, image retrieval should be performed, for a given K, with different random chosen initial partitions, and the average performance is measured.

2.5 Sparse representation

Sparse representation models data vectors with a linear combination of a small number of basis elements. Often, elements are chosen from a so called over-complete dictionary, which is a collection of elements such that the number of elements exceeds the dimension of the elements. This theory has been used in machine learning, neuroscience, signal processing, pattern recognition and achieved state-of-the-art results [48]. In this thesis, it will be used for generating the histogram and the related fundamentals will be introduced in this section.

2.5.1 Definition

In the point of view of matrix factorization, for a given data matrix $\mathbf{X} \in \mathbb{R}^{J\times N}$, we want to find a basis matrix (dictionary) $\mathbf{D} \in \mathbb{R}^{J\times K}$ and a coefficient matrix $\mathbf{C} \in \mathbb{R}^{K\times N}$ that can represent the original data matrix $\mathbf{X} \in \mathbb{R}^{J\times N}$:

$$\mathbf{X} \approx \mathbf{DC} \tag{2.49}$$

where the columns of $\mathbf{X}$ are vectors to be represented and columns of $\mathbf{D}$ are basis vectors and rows of $\mathbf{C}$ consist of the coefficients by which a vector can be represented with a linear combination of basis vectors, J represents the dimension of the vector, K represents the number of vectors in the dictionary, N indicates the number of vectors in the data matrix.

This problem can be expressed as looking for $\mathbf{D}$ and $\mathbf{C}$ that can minimize the reconstruction error :

$$\arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X} - \mathbf{DC}\|_F$$

where $\| * \|_F$ represents the error function.

Different constrains imposed on basis matrix $\mathbf{D}$ and the coefficient matrix $\mathbf{C}$ will lead to different methods of representations. In Vector Quantization (VQ), each row of $\mathbf{C}$ is constrained to be a unary vector, with one element equal to one and the other elements equal to zero. In other words, every vector is approximated by a single basis vector. In Principle Components Analysis (PCA), it constrains the columns of $\mathbf{D}$ to be orthogonal and the rows of $\mathbf{C}$ to be orthogonal to each other, which allows a distributed representation in which each vector is approximated by a linear combination of all the basis vectors. And the constraints of Non-negative Matrix Factorization (NMF) [49] are the non-negativity on both $\mathbf{D}$ and $\mathbf{C}$:

$$\begin{aligned} &\arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X} - \mathbf{DC}\|_F \\ &s.t.\ \mathbf{D} \succeq 0, \mathbf{C} \succeq 0. \end{aligned} \tag{2.50}$$

These non-negativity constraints permit the combination of multiple basis vectors to represent a vector. But only additive combinations are allowed, because the elements of $\mathbf{D}$ and $\mathbf{C}$ are all positive. Another most useful properties of NMF is that it usually produces a sparse representation, which means that the number of non-zeros elements in each row of $\mathbf{C}$ is much less than the dimension of the row. However, this sparseness cannot be controlled. To solve this problem, Equation 2.50 can be extended to *Lasso* problem [50] with positive constraints, which provides a sparse solution:

$$\begin{aligned} &\arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X} - \mathbf{DC}\|_{\ell_2} + \lambda\|\mathbf{C}\|_{\ell_1} \\ &s.t.\ \mathbf{D} \succeq 0, \mathbf{C} \succeq 0. \end{aligned} \tag{2.51}$$

Parameter λ controls trade-off between accuracy and sparseness. When $\lambda = 0$, this equation is equivalent to NMF.

In this context, given a training data matrix $\mathbf{X_T}$, we can find a dictionary $\mathbf{D_T}$ by different dictionary learning methods [51]. And then given a test data matrix $\mathbf{X}$, the vector $\mathbf{x}$ in the test data matrix can be represented by a linear combination of a few basis vectors from the dictionary $\mathbf{D_T}$ and the coefficients in each row of $\mathbf{C}$ can be seen as the weight parameters of basis vectors. This step is solved by the LARS algorithm [52] provided by the toolbox SPAMS [53].

2.5.2 Sparse representation based histogram

The sparse representation based histogram of data matrix $\mathbf{X}$ is proposed to be defined as:

$$h_j = \sum_{i=1}^{N} C_{ij} \tag{2.52}$$

where $C_i \in \mathbb{R}^{K\times 1}$ is the row of $\mathbf{C}$ and h_j indicates the value of the j-th bin of the histogram, $j = \{1, 2, \ldots, K\}$. In this way, the values of bins represent the total weight of corresponding basis vectors in the sparse representation of data matrix.

For easy understanding, considering there are 9 vectors $X_1, X_2, \ldots, X_9$ in a vector space that are divided into 5 partitions. These 5 partitions are represented by 5 cluster centers $D_1, D_2, \ldots, D_5$ as shown in Figure 2.8. According to the classical definition of histogram, in which the value of bins is the number of vectors that fall into corresponding partitions of vectors, the histogram of these vectors is shown in Figure 2.9(a). In other words, if we see the centers of partitions as basis vectors, target vector is represented only by one basis vector.

Let's consider the sparse representation of these 9 vectors $\mathbf{X} = [X_1\ X_2\ \cdots\ X_9]$ with 5

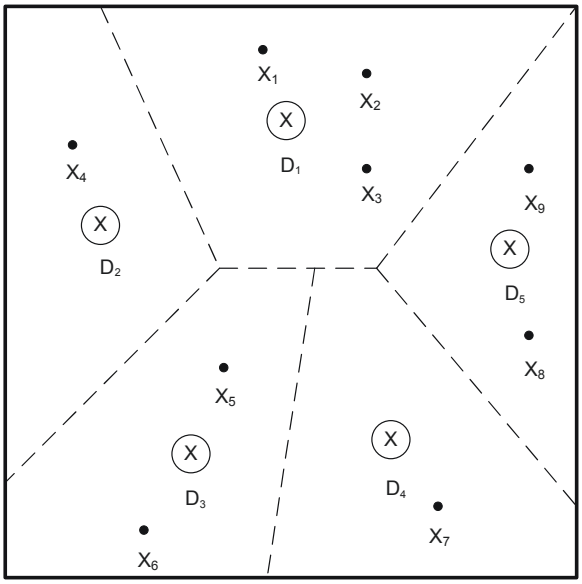


Figure 2.8: Vectors in a vector space

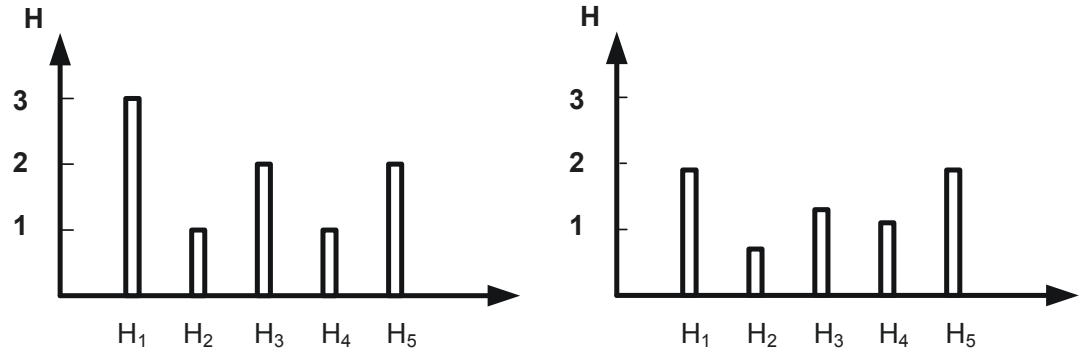
cluster centers as basis vectors $\mathbf{D} = [D_1\ D_2\ \cdots\ D_5]$. And assuming $\mathbf{C}$:

$$\mathbf{C} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{pmatrix} = \begin{pmatrix} 0.5 & 0.2 & 0 & 0 & 0 \\ 0.6 & 0 & 0 & 0 & 0.1 \\ 0.7 & 0 & 0 & 0.05 & 0.1 \\ 0.1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.3 & 0 \\ 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0.1 \\ 0 & 0 & 0 & 0.1 & 0.8 \\ 0 & 0 & 0 & 0 & 0.8 \end{pmatrix} \tag{2.53}$$

So according to Equation 2.52, the sparse representation based histogram of these vectors is got and as shown in Figure 2.9(b):

$$H = \{h_j\} = \{1.9\ \ 0.7\ \ 1.3\ \ 1.15\ \ 1.9\} \tag{2.54}$$

Different from classical histogram, in sparse representation based histogram, one vector is represented not only by one basis vector but by a few basis vectors. This will provide more information about the relations between one vector and other vectors in the vector space. It means that every vector is sparsely represented by basis vectors with different



(a) Classical histogram (b) Sparse representation based histogram

Figure 2.9: Comparison of two histograms

weight, as shown in follows:

$$
\begin{aligned}
X_1 &\approx 0.5D_1 + 0.2D_2 \\
X_2 &\approx 0.6D_1 + 0.1D_5 \\
X_3 &\approx 0.7D_1 + 0.05D_4 + 0.1D_5 \\
X_4 &\approx 0.1D_1 + 0.5D_2 \\
X_5 &\approx 0.4D_3 + 0.3D_4 \\
X_6 &\approx 0.9D_3 \\
X_7 &\approx 0.7D_4 + 0.1D_5 \\
X_8 &\approx 0.1D_4 + 0.8D_5 \\
X_9 &\approx 0.8D_5
\end{aligned}
\tag{2.55}
$$

2.6 Similarity measurement

The similarity between two images can be measured by the distances between feature descriptors of images. A similarity measure assigns a lower distance or a higher score to the most similar images. As we choose histogram as feature descriptors, in this section, we detail the similarity measurement between two histograms H_Q and H_D.

Reviews of similarity measurement can be found in [41,54]. In general, similarity measurements between histograms could be classified into two groups: bin-to-bin approaches and cross-bin approaches. The bin-to-bin similarity measurements compare contents of

corresponding histogram bins, that is, they compare $H_Q(i)$ and $H_D(i)$ for i, but not $H_Q(i)$ and $H_D(j)$ for $i \neq j$. The cross-bin measures also compare non-corresponding bins. Cross-bin distances make use of the ground distance concept defined as the distance between the representative features for bin i and bin j.

Some terms used in this section need to be defined firstly.

Metric space: A space $\mathbb{R}^N$ is a metric space if for any of its two elements x and y, there exists a distance $d(x, y)$, that satisfies the following prosperities:

- $d(x, y) \geq 0$ (non-negativity)
- $d(x, y) = 0$ if and only if $x = y$ (identity)
- $d(x, y) = d(y, x)$ (symmetry)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Partial Matching: Similarity is partially obtained when the number of bins of one histogram is smaller than that of the other. The similarity is measured only with respect to the most similar part of the larger histogram.

2.6.1 Bin-to-bin similarity measurements

This category of measurements compare the corresponding bins of two histograms H_Q and H_D (both have N bins). The similarity between two histograms is a combination of these bin-to-bin comparisons.

Minkowski-form distance

The well known Minkowski-form distances are defined by:

$$d_{L_p}(H_Q, H_D) = \Big(\sum_{i=1}^{N} |H_Q(i) - H_D(i)|^p \Big)^{1/p} \tag{2.56}$$

When $p = 1, 2$ *or* ∞, these distances are also referred as Manhattan distance, Euclidean distance or Chebyshev distance respectively. These three are most common used in image retrieval.

Histogram intersection

Histogram intersection [55] is defined as:

$$d_{\cap}(H_Q, H_D) = \sum_{i=1}^{N} \min(H_Q(i), H_D(i)) \tag{2.57}$$

It assigns a higher score when two histograms are more similar. And it is attractive because of its ability to handle partial matches and low complexity as only minimum and addition operations are required.

Chi-squared distance

The χ^2 distance is defined as:

$$d_{\chi^2}(H_Q, H_D) = \sum_{i=1}^{N} \frac{\big(H_Q(i) - H_D(i)\big)^2}{H_Q(i) + H_D(j)} \tag{2.58}$$

Note that χ^2 distance does not have the triangle inequality prosperity.

Kullback-Leibler divergence distance

The Kullback-Leibler (K-L) divergence distance [56] is defined as:

$$d_{KL}(H_Q, H_D) = \sum_{i=1}^{N} H_Q(i) \log \frac{H_Q(i)}{H_D(i)} \tag{2.59}$$

From the point of view of information theory, the K-L divergence distance measures how inefficient on average it would be to code one histogram using the other as the code-book. However, the Kullback-Leibler divergence is not symmetric and is numerically unstable. To overcome these problems, Jeffrey Divergence distance is proposed.

Jeffrey divergence distance

The Jeffrey divergence (JD) distance is defined as:

$$d_{JD}(H_Q, H_D) = \sum_{i=1}^{N} \Big(H_Q(i) \log \frac{H_Q(i)}{H_D(i)} + H_D(i) \log \frac{H_D(i)}{H_Q(i)} \Big) \tag{2.60}$$

Conclusion of bin-to-bin similarity measurement

These similarity measurements are appropriate in different areas. For example, the Kullback-Leibler divergence is justified by information theory and the χ^2 distance by statistics. The advantage is that they are easy to compute with low complexity. The drawback is that they only consider the corresponding bins and neglect the information across bins. Moreover, they are sensitive to the choice of bins.

2.6.2 Cross-bin similarity measurements

All previous measures perform the comparison by considering the differences between corresponding histogram bins only. As the traditional histogram has discontinuities at the bin boundaries, a small shift in the feature value can result in an assignment to neighboring bins instead of the original one, for example, H_1 and H_2, as shown in Figure 2.10. Of course it would be desirable to consider these two histograms closer to each other than to H_3. The similarities observed from the previous measurements would find no similarity between H_1 and H_2 neither between H_1 and H_3. So to compare these histograms, other distances should be presented.

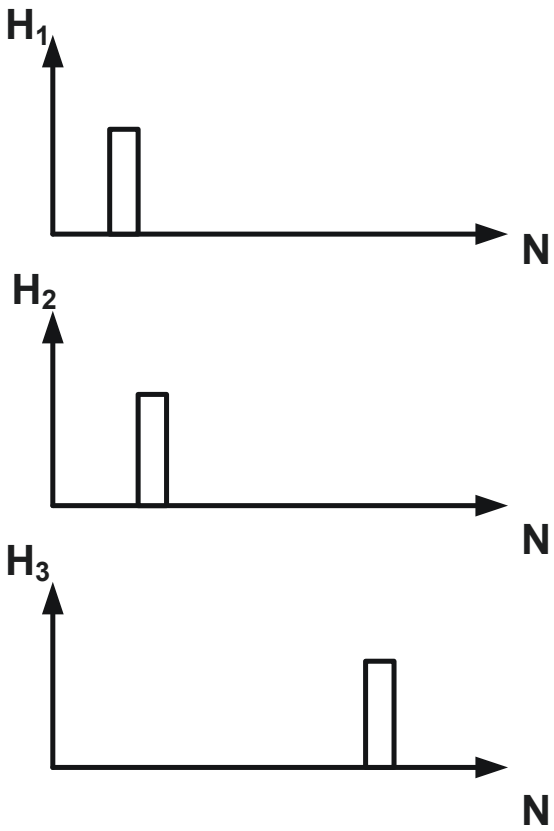


Figure 2.10: Illustration of histograms giving the motivation of ground distance measures

Quadratic-form distance

Considering similarity between bins, quadratic-form (QF) distance [57] is defined as:

$$d_{QF}(H_Q, H_D) = \sqrt{(H_Q - H_D) \cdot A \cdot (H_Q - H_D)^T} \tag{2.61}$$

where $A = [a_{ij}]$ is a similarity matrix, and a_{ij} denotes ground distance between bins $H_Q(i)$ and $H_D(j)$. Common selections for A are

$$a_{ij} = 1 - \frac{d_{ij}}{d_{\max}} \tag{2.62}$$

or

$$a_{ij} = \exp\Big(- \beta\Big(\frac{d_{ij}}{d_{\max}}\Big)^2\Big) \tag{2.63}$$

where β is a positive constant, with d_{ij} being the Euclidean distance between bins $H_Q(i)$ and $H_D(j)$, $d_{\max}$ being maximum of d_{ij}.

Earth movers distance

The Earth movers distance (EMD) [58] compares histogram matching with a transportation problem, whose purpose is to minimize the cost of shipping goods from one location to another so that the needs of each arrival area are met and every transportation operates within its limited capacity. Given two histograms, one can be seen as the hill of earth and the other as holes in the ground. Then the EMD measures the least amount of work needed to fill the holes with earth from hills. These are represented by the ground distance matrix $D = [d_{mn}]$ where d_{mn} is the ground distance between bin m and bin n. In general, ground distance can be any distance and will be chosen according to the problem at hand [59]. A flow $F = [f_{ij}]$ need to be found, with f_{ij} the flow between bin i and bin j, which can minimize the overall work:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij} f_{ij} \tag{2.64}$$

where f_{ij} is subject to the following constraints:

1. Earth is moved only in one direction:

$$f_{ij} \geq 0 \qquad \forall i, j \tag{2.65}$$

2. Earth can not be moved more than available:

$$\sum_{j=1}^{N} f_{ij} \leq H_Q(i) \qquad \forall i \tag{2.66}$$

3. Earth can not be moved more than required:

$$\sum_{i=1}^{N} f_{ij} \leq H_D(j) \qquad \forall j \tag{2.67}$$

4. The total flow equals the minimum amount of earth available in H_Q or earth required in H_D:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} = \min\left(\sum_{i=1}^{N} H_Q(i), \sum_{j=1}^{N} H_D(j)\right) \tag{2.68}$$

The EMD is then defined as the resulting work normalized by the total flow:

$$d_{EMD}(H_Q, H_D) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij} f_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}} \tag{2.69}$$

Note that EMD also supports partial matches, similar as histogram intersection. EMD is even more suitable for histogram comparison than the quadratic form as it does not only consider ground distance but also pays attention to what bins have already been used in the comparison. However, computational complexity is rather high [41].

2.6.3 Conclusion on similarity measurements

Table 2.1 which is partially taken from [54] provides the comparison of main properties of similarity measurements aforementioned. 'Y/N' means that the property occurs only in special case. From this table, we can conclude that all the measurements are symmetric except K-L divergence distance; only L_p distance holds triangle inequality, furthermore, histogram intersection and EMD could solve the problem of partial matching. From the point of view of computing load , bin-to-bin measurements have lower complexity than cross-bin measurements.

Table 2.1: Properties of different similarity measurements

Properties	L_p	$\bigcap$	Chi-Squared	KL	JD	QF	EMD
Symmetry	Yes	No	Yes	No	Yes	Yes	Yes
Triangle inequality	Yes	Y/N	No	No	No	Y/N	Y/N
Partial matching	No	Yes	No	No	No	No	Yes
Complexity	Low	Low	Low	Low	Low	High	High

Generally speaking, triangle inequality and partial matching are not obligatory in color texture retrieval, but symmetry and lower complexity are always preferred. From above

considerations, we will choose bin-to-bin measurements, in particular, L_1 distance and χ^2 distance as similarity measurement adopted in the proposals presented in following chapters.

2.7 Performance evaluation

To evaluate and compare the performance of different CBIR algorithms, an objective performance evaluation is necessary. Providing a clear and broadly understood performance evaluation allows researchers to more fully understand the strengths and limitations of their algorithms and to compare their results with other algorithms objectively. In this section, we introduce some common performance evaluation measurements.

2.7.1 Precision and recall

The most common evaluation measurement used in CBIR is precision and recall pair. For each query image, system returns a ranked list of images. Each image in the list is determined as either relevant or not to the query. Then the performance can be evaluated by precision and recall. Precision indicates the retrieval accuracy and is defined as the ratio of the number of relevant retrieved images over the number of total retrieved images. Recall indicates the ability of retrieving relevant images from the database. It is defined as the ratio of the number of relevant retrieved images over the total number of relevant images in the database:

$$\begin{aligned} Precision &= \frac{\sharp(relevant \quad retrieved \quad images)}{\sharp(retrieved \quad images)} \\ Recall &= \frac{\sharp(relevant \quad retrieved \quad images)}{\sharp(relevant \quad images \quad in \quad database)} \end{aligned} \tag{2.70}$$

$\sharp(\alpha)$ denotes cardinality of set α.

In practice, if we assume q is the number of relevant retrieved images , s is the number of non-relevant retrieved images , t is the number of relevant images not retrieved in a

database, then precision and recall pair could be expressed as:

$$\begin{aligned} Precision &= \frac{q}{q+s} \\ Recall &= \frac{q}{q+t} \end{aligned} \tag{2.71}$$

so $0 \leq Recall \leq 1$, $0 \leq Precision \leq 1$.

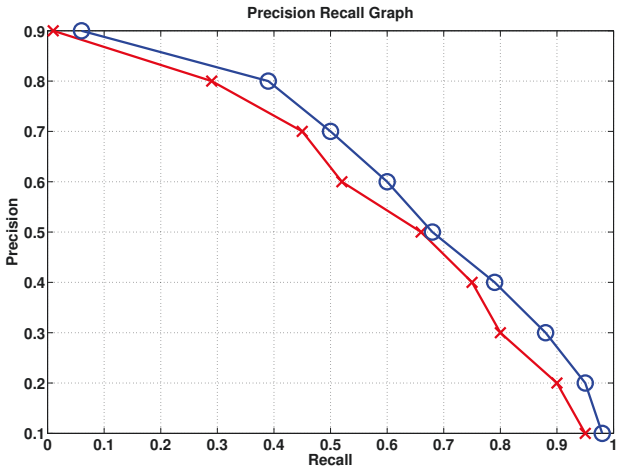


Figure 2.11: Precision Recall Graph

The precision normally decreases while recall increases. This is because in the process of trying to retrieve all images relevant to a query, certain non-relevant images are also retrieved. Thus traditional using of precision and recall pair usually presented as "Precision vs Recall graph" to demonstrate the performance of a system. One example is shown in Figure 2.11. In this graph, the line with "o" (blue one) has a better performance than the one with "+" (red one) as when these two have the same level of recall (ability of retrieving relevant images from database), the blue one has a higher precision (retrieval accuracy).

2.7.2 Average retrieval rate

Average retrieval rate (ARR) [60] is often used in the literatures about texture retrieval. For a given query image, the retrieved images are ordered according to increasing dissimilarity with the given query, with the top retrieved image being most similar to the query. The retrieval rate (RR) for this query is defined as the percentage of the number of

correct retrieved images over the total number of relevant images in the database in the top K retrieved images:

$$RR = \frac{\sharp(relevant \quad retrieved \quad images)}{\sharp(relevant \quad images \quad in \quad database)} \tag{2.72}$$

ARR is defined as the mean value of the set of retrieval rate in top K retrieved images for each query. Obviously, ARR is related to the number of retrieved images. So it is possible to construct receiver operating characteristic (ROC) curves by plotting ARR against K. This allows to study the retrieval performance as the number of retrieved images inscrease. So with ARR, we have two ways to compare the performances of different approaches. One way is to compare ARRs with a given K: the higher the better. And the other way is to compare the ROC curves of ARR: a ROC curve of an approach lying above the ROC curve of another approach demonstrates the increase of performance, as shown in Figure 2.12: approach 2 outperforms approach 1.

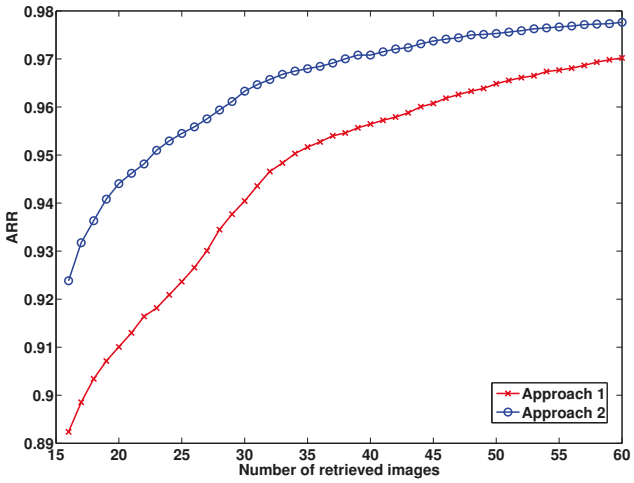


Figure 2.12: ARR according to the number of top K retrieved images

2.7.3 Average of normalized modified retrieval rank

The average of normalized modified retrieval rank (ANMRR) is a performance evaluation used by MPEG-7 to evaluate the performance of a retrieval system [61]. This evaluation combines the precision and recall pair to obtain a single objective value, in which queries and sets of ground truth images are chosen manually. For each query a set of ground truth images are most relevant images to the query and the relevant images in

ground truth set are not ordered in any way. A good algorithm is expected to retrieve all ground truth images for a given query image.

Assume $N_G(Q)$ is the number of relevant images in ground truth set for a given query Q and M is the maximum number of relevant images in ground truth set for all Q queries, it means that:

$$M = \max(N_G(Q_1), N_G(Q_2), \ldots, N_G(Q_Q)) \tag{2.73}$$

Then for a given query Q, each relevant image k is assigned a rank value $rank(k)$ that is equivalent to its rank in the retrieved relevant images if it is in the first $K(Q) = \min[4 \times N_G(Q), 2 \times M]$ query results; or a rank value $K(Q) + 1$ if it is not. The average rank $AVR(Q)$ for query Q is computed as:

$$AVR(Q) = \sum_{k=1}^{N_G(Q)} \frac{rank(k)}{N_G(Q)} \tag{2.74}$$

The modified retrieval rank $MRR(q)$ is computed as:

$$MRR(Q) = AVR(Q) - 0.5 - 0.5 \times N_G(Q) \tag{2.75}$$

$MRR(Q)$ takes value 0 when all the relevant images are within the first $K(Q)$ retrieved results.

Then the normalized modified retrieval rank $NMRR(Q)$ is computed as:

$$NMRR(Q) = \frac{MRR(Q)}{K(Q) + 0.5 - 0.5 \times N_G(Q)} \tag{2.76}$$

The $NMRR$ is in the range of $[0, 1]$ and smaller values represent a better retrieval performance. For example, suppose that for a query Q, the relevant images in ground truth set are $I_1, I_2, \ldots, I_{10}$, so $N_G(Q) = 10$. The ideal result is that the top 10 retrievals are these relevant images $I_1, I_2, \ldots, I_{10}$, then their retrieval ranks are $rank(I_1) = 1, rank(I_2) = 2, \ldots, rank(I_{10}) = 10$ respectively. So NMRR for this query Q is 0.

ANMRR is defined as the average NMRR over a set of queries:

$$ANMRR = \frac{1}{N} \sum_{Q=1}^{N} NMRR(Q) \tag{2.77}$$

where NQ is the number of query images.

2.7.4 Equal error rate

Equal Error Rate (EER) [62] is often used to evaluate the performance of face recognition algorithm. When recognition is performed, similarity between images must be observed. Images are considered as similar if the distance between their features descriptors is under a given threshold. So considering a query image belonging to class A, two things could occur: on one hand, it could be recognized rightly; on the other hand, it could be falsely rejected from class A, then the ratio of how many images of class A are in this situation is called False Rejected Rate (FRR). In contrast, considering a query image out of class A, when it is compared with the images of class A, it could be rejected rightly or it could be falsely accepted as class A, then the ratio of how many images of other class are in this situation is defined as False Accept Rate (FAR). These two rates will change when the threshold change. When FRR and FAR take equal values, an equal error rate (EER) is got. The lower the EER is, the better is the performance of system, as the total error rate is the sum of FAR and FRR. One example of EER is shown in Figure 2.13.

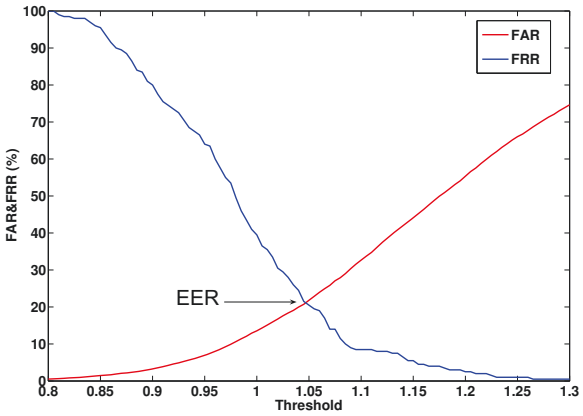


Figure 2.13: Equal Error Rate

2.7.5 Choice of performance evaluation

In this thesis, we will choose precision and recall pair because of their classic and ARR in color texture retrieval experiments and EER in face recognition experiments because they are commonly used in other literatures to evaluate the performance of approaches and compare them with other reported methods, including state-of-the-art methods.

2.8 Conclusion

In this chapter, some fundamental concepts concerned with our approaches for CBIR have been introduced.

First, we have introduced the two commonly used transforms: DCT and DWT, which are used to transfer the images to transform domain. The properties of transform coefficients are analyzed, from which we give theoretical bases for generating feature vectors directly from transform coefficients. Furthermore, the method for extracting 4×4 block DCT coefficients directly from 8×8 block DCT coefficients is proposed.

Histogram of feature vectors is chosen as the descriptor for image retrieval. So the concepts of histogram have been presented.

Furthermore, K-means and sparse representation used for generating histograms have been introduced. And sparse representation based histogram is proposed. Different from classical histogram, sparse representation based histogram provides more information on the relations between a vector and its related basis vectors.

Once the descriptor of images is ready, it should be considered how to measure the similarity between them . So we have presented the common used similarity measurements in CBIR, from which two kinds of distances: Manhattan distance and chi-squared distance are chosen for our approaches because of their lower computing load.

Finally, the performance evaluation for CBIR especially for face recognition and texture retrieval have been presented, from which precision and recall pair, ARR and EER are chosen as they are widely used by other researches in this field.

All aforementioned concepts allow constructing the base of image retrieval in transform domain, including feature extraction, feature representation, similarity measurement and performance evaluation. In the following chapters, approaches of CBIR in DCT domain and DWT domain are proposed.

3 Image descriptors in DCT domain

3.1 Introduction

Image retrieval in transform domain has widely been studied since majority of the images are stored in compressed format and most of compression technologies adopt different kinds of transforms to achieve compression. Compared to traditional approaches of indexing compressed images which need to decode the images to the pixel domain first, the approaches working directly in transform domain lead to low computational load and storage requirements.

As a transform adopted in JPEG compression standard, DCT has demonstrated to be a powerful tool to extract proper features from images. In this chapter, supplement to some improvements compared to an existing approach for face recognition using block DCT transform, two novel approaches are proposed, which are also extended for texture retrieval. Finally, with the proposal of color features, new approaches are applied in color texture image retrieval by combination of color and texture features.

The rest of the chapter is organized as follows: related works are introduced firstly, followed by the improvements given in Section 3.3. Then a novel approach for face recognition and texture retrieval on gray-scale images is presented in Section 3.4. A proposal for color texture image retrieval by combination of color and texture features is then presented in Section 3.5.

3.2 Related works

This section is divided into two parts: first, we give the general description of related works on face recognition and image retrieval in DCT domain and then detail an existing

approach based on the histogram of DCT blocks on which we make improvements, and this also gives us the basic framework of our two proposals.

3.2.1 Face recognition and image retrieval in DCT domain

Numerous researches on face recognition and image retrieval use the DCT to extract features from images.

In the context of face recognition, Hafed has proposed an approach that computes the DCT on the entire normalized image and retained low-to-mid frequency DCT coefficients as feature vectors [63]. Ramasubramanian has presented an approach based on a combination of DCT, principal component analysis (PCA) and the characteristics of the Human Visual System, in which only low-frequency coefficients are selected for feature vectors and PCA is employed to select a basis set of features called cosine-faces [64]. Another selection of DCT coefficients based on a data-dependent approach, discrimination power analysis, which is used to find coefficients that have the strong ability to discriminate various classes has been published in [65].

Since block DCT transform is widely used in image and video compression, in order to establish a direct link with compressed images, many researcher have presented approaches for face recognition based on block transform. For example, Shneier has proposed to use the average value of DCT coefficients in each 8×8 block as feature vectors [66], and Nefian has selected coefficients from each DCT block to train Hidden Markov Model (HMM) for face recognition [67]. Eickeler has used the first fifteen coefficients of 8×8 DCT block to train 2-D HMM for face recognition [68]. Zhong has constructed the feature descriptors based on the histogram of DCT blocks and these descriptors are used to do face recognition [69,70].

In the context of image retrieval, color and texture are two important features that are used in CBIR. In [71], the statistical information of DCT coefficients has been used as texture features. As DCT has the high capability of energy compaction, in [72], the upper left DCT coefficients transformed from the entier image are chosen to form the feature vector of an image and these selected coefficients are categorized into 4 groups: one is DC coefficients and other three includes the coefficients which have vertical, horizontal and diagonal information.

A combined use of color and texture would provide better performance than using color or texture alone. In general, from this previous consideration, those approaches could be divided into two groups: jointly and separately [29].

Under the jointly aspect, in [73], images were firstly converted to YCbCr color space and then transformed by 8×8 block DCT. Feature vector consisted of the first 36 coefficients of each block from Y channel I_Y and DC coefficient of each block from Cb channel I_{Cb} and Cr channel I_{Cb}.

Another aspect of analyzing color images is to process color and texture separately, that means to transform the color image into luminance and chrominance components and then, extract color feature and texture feature separately. For example, in [74], images are converted to YCbCr color space and then each component is transformed by 8×8 block DCT. The values of DC coefficients of I_Y, I_{Cb} and I_{Cr} in each block are used directly as color feature and the mean value and standard deviation of energies of AC coefficients in each block from Y channel are used as texture feature. In [75], DC coefficients from each 8×8 block are grouped to a DC image, the color histogram of this image obtained by summing up the number of pixels with similar values in HSV color space is used as the color feature. And the variance and expectation of first 9 AC coefficients of each block are used as texture feature. In [76] [77], features were also extracted from 8×8 block DCT transform in YCbCr color space. DC coefficients of each block are extracted from I_Y, I_{Cb} and I_{Cr} in order to form a 3D vector which is treated as color feature and the AC coefficients of each block inside each diagonal line of zig-zag scan constructed texture feature. In [78], the average values of all 4×4 sub blocks in a 8×8 block from each component in YCbCr color space are used as color feature and the mean and standard deviation of the sum of 6 groupes of selected coefficients in one 8×8 block are used as texture feature.

3.2.2 Image retrieval based on histogram of DCT blocks

In [70] and [79], authors have proposed to use the histogram of quantized DCT blocks for image retrieval, which is the basic framework of our approaches. Images are firstly decomposed by 4×4 block DCT. As a same scene taken at different luminance level will lead to different DCT blocks, to normalize the luminance, preprocessing steps are done before extracting feature vectors. This is done by rescalling the DCT coefficients according to the average luminance level, which is calculated based on the DC coefficients of the DCT blocks.

Assume there are N DCT blocks in an image i, and the DC value for each block is denoted by $DC_j(i), 1 \leq j \leq N$. From these DC values, we can calculate the mean DC

value for this image:

$$DC_{mean}(i) = \frac{1}{N}\sum_{j=1}^{N} DC_j(i) \tag{3.1}$$

Then the average luminance $DC_{allmean}$ of all images in database is calculated:

$$DC_{allmean} = \frac{1}{M}\sum_{i=1}^{M} DC_{mean}(i) \tag{3.2}$$

where M denotes the total number of images in the database.

Then the ratio of luminance rescalling for image i is calculated as:

$$R_i = \frac{DC_{allmean}}{DC_{mean}(i)} \tag{3.3}$$

And finally all the DCT coefficients of image i are normalized, with respect to R_i, by rescalling them:

$$\overline{DCT_i} = DCT_i \times R_i \tag{3.4}$$

where DCT_i is the DCT coefficients of image i.

After this normalization, the DCT coefficients are quantized by a quantization parameter QP:

$$\overline{\overline{DCT_i}} = \frac{\overline{DCT_i}}{QP} \tag{3.5}$$

In this approach, a DCT block without DC coefficient will be defined as AC-Pattern. So after preprocessing, AC coefficients from each 4x4 DCT block which are ordered with left-to-right and top-to-bottom way are used to construct AC-Patterns as illustrated in Figure 3.1 (Duplicated from [79]). The zero value at the end of the AC-Pattern are skipped which can reduce the size of vector. From now we named this method of constructing AC-Pattern as *Linear scan*.

DC-Pattern is defined as the directions that have largest differences between DC value of current block and DC values of neighboring blocks. The process of forming DC-Pattern is shown in Figure 3.2 (Duplicated from [70]). Eight differences between DC coefficient of the current block and its 8 neighbors are calculated. The ninth difference is the difference between current DC value and the mean of all the nine neighboring DC values. The absolute value of these differences are ordered in descending order and the first γ direction-values with largest differences are taken to form DC-Pattern. Here γ is a parameter which can be adjusted for a better retrieval performance.

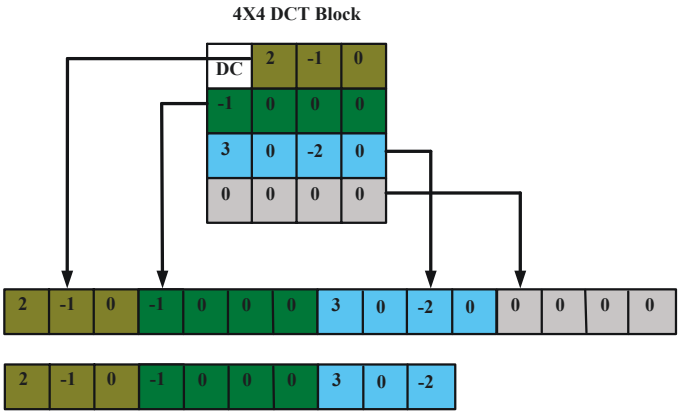


Figure 3.1: Forming AC-Pattern

Histogram of patterns defined as the occurrence of patterns in image is adopted as feature descriptors used for image retrieval.

Based on experiments, it appears that only a restricted part of patterns exists with high occurrence and a large number of patterns appears rarely, so the histograms of most frequent patterns (AC-Pattern and DC-Pattern) are used for image retrieval in which Manhattan distance is used to measure the similarity.

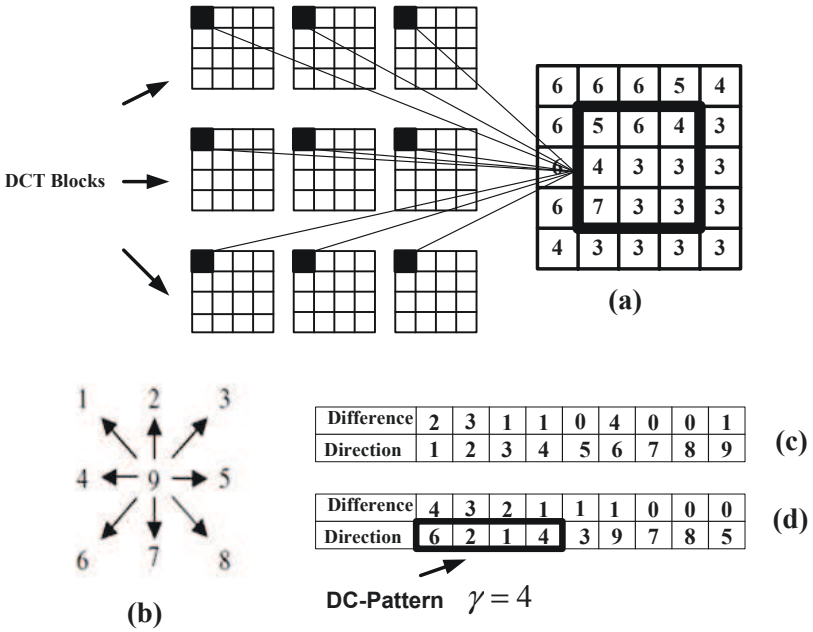
Dimension of patterns or descriptors affect the computational load and two aspects could be improved in this method: the first one is the dimension of AC-Pattern. Although the zero values at the end of the AC-Pattern are skipped, the max dimension of AC-Pattern is still 15 and this can be reduced; the second one is the method for histogram generation, which aims to reduce the number of bins of the histogram.

3.3 Improvements on linear scan method

In this section, we detail the improvements on *Linear scan* method for constructing AC-Patterns. A general description is introduced firstly, and then improvements on AC-Pattern construction and histogram generation are given. Finally, the experimental results of improved approach are compared with *Linear scan* method.

3.3.1 General Description

As we told in the previous section, we mainly focus on improving AC-Pattern construction and histogram generation. Concerning AC-Pattern construction, we propose to use a new way to construct AC-Patterns in order to reduce their dimensions. In the step



(a) DC coefficients are extracted from neighboring blocks (b) Definition of directions (c) Differences for each direction (d) Directions with largest differences form DC-Patterns

Figure 3.2: DC-Pattern construction

of histogram generation, we merge adjacent patterns. We named this improved approach for AC-Pattern construction and histogram generation as *Zigzag-Pattern*.

The flowchart of image retrieval process is shown as Figure 3.3. 4×4 block DCT transform is used as in *Linear scan*. So for each block we get 1 DC coefficient and 15 AC coefficients. AC-Pattern is referred as layout of an arrangement of AC coefficients in one DCT block, therefore, the total number of AC coefficients in a DCT block is 15, but the number of coefficients that are used to construct the AC-Pattern can be adjusted. Time consuming and performance can change because of this adjustment. The descriptors are constructed from histograms of AC-Patterns (H_{AC})and histograms of DC-Patterns (H_{DC}). Manhattan distance is used to measure the similarity between descriptors of query and images in the database.

3.3.2 Preprocessing

To reduce the impact of luminance variation, preprocessing steps as detailed in Section 3.2.2 need to be done before generating AC-Patterns and DC-Patterns.

As QP defines the sensitivity for the observation of coefficients values, from Equation 3.5, we can observe that high QP truncates coefficients leading to zero values, which

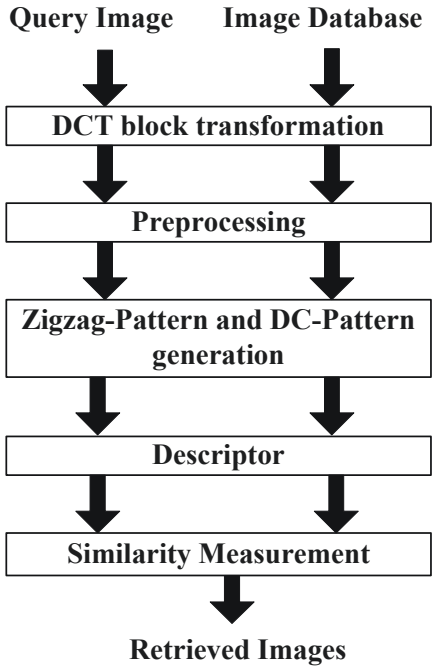


Figure 3.3: Flowchart of image retrieval

means if QP is rather high, only a small number of different AC-Patterns will be left, even all the coefficients could be zero. This will decrease the performance of image retrieval obviously. In contrast, if this value is low, the total number of different AC-Patterns will be very high, that will make the processing histogram generation more time-consuming and complicated. So there will be a trade-off value of quantization parameter between performance and time efficiency.

3.3.3 Construction of AC-Pattern histogram

There are two methods of scanning for arranging the coefficients in AC-Pattern. The first way is a row-by-row manner, as used in *Linear scan* method. In this way, AC coefficients are ordered from left-to-right and top-to-bottom. The second way is zigzag scan as defined in JPEG standard, which we propose to use. For most images, most of the signal energy lies at low frequencies coefficients and these appear in the upper left corner of the DCT. Using zigzag scan, the coefficients are in the order of increasing frequency. So comparing with linear scan, zigzag scan gains more advantages in using coefficients that have higher energies in the condition of limited number of coefficients used. These two methods are shown in Figure 3.4.

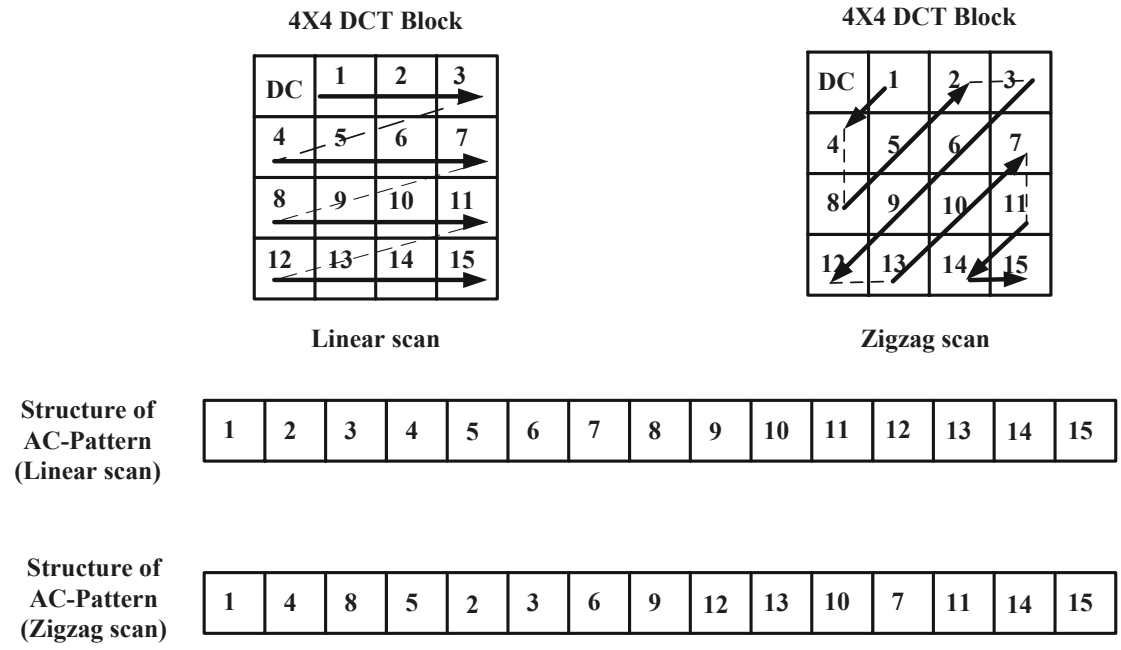


Figure 3.4: Linear and zigzag scan

The number of coefficients (Nc) used in the AC-Pattern should be considered further. When this value is large, more high-frequency coefficients are included in the AC-Patterns. It means that the retrieval performance will be more sensitive simultaneously to thin details with high frequency contents and also to the noise. Furthermore, the total number of different AC-Patterns will also be larger and that will lead to more time-consuming in image retrieval. When this value is small, the total number of different AC-Patterns will be small too. Although it leads to less time-consuming, it will decrease the performance of the retrieval too. We will have to find an optimal value of Nc for best performance of retrieval and less time-consuming.

Since observed AC-Patterns are numerous, we have the objective of reducing their number. As differences between patterns can be weak, issued from small differences between their coefficients, patterns can be considered quite similar and we will call them adjacent patterns, which will be merged by observing distances between coefficients in AC-Patterns, as shown in Figure 3.5.

Adjacent patterns will correspond to blocks whose frequency contents are close in one or serval frequencies. AC-Pattern i and j are said to be adjacent patterns if:

$$|C_i(1) - C_j(1)| \leq Th \textbf{ or } |C_i(2) - C_j(2)| \leq Th \textbf{ or} \cdots \textbf{ or } |C_i(m) - C_j(m)| \leq Th \quad (3.6)$$

where $C_i(k)$ and $C_j(k)$ $(1 \leq k \leq m)$, represent AC coefficients in AC-Patterns i and j, Th is the threshold, m indicates the number of coefficients in AC-Pattern. When two patterns meet this requirement, they will be merged. In our proposal, $Th = 1$. So we tolerate a difference of 1 on each coefficient value inside the pattern.

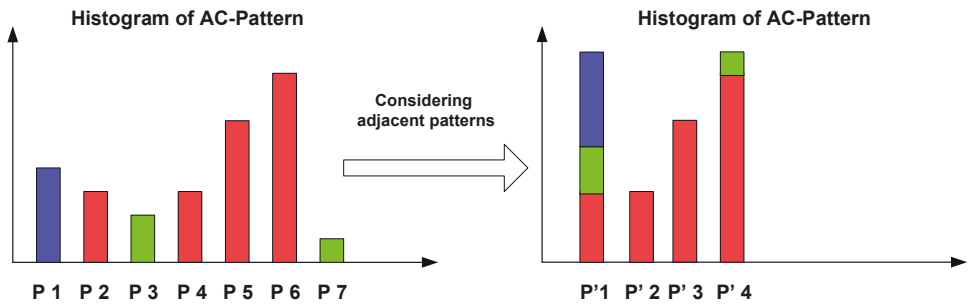


Figure 3.5: Merging adjacent patterns in histogram

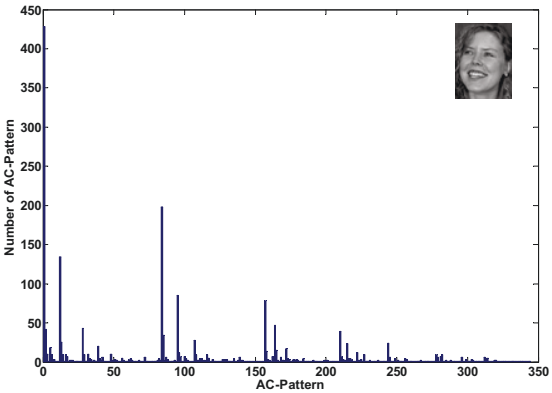
Furthermore, there is one special AC-Pattern in which all the AC-coefficients are zero and this AC-Pattern mainly corresponds to uniform blocks of image. We exclude this pattern from H_{AC}. To find and merge the adjacent patterns, the histogram of AC-Pattern of the database is generated and the bins are arranged in adjacent order first and then the adjacent patterns that should be merged are found. Finally, with these adjacent patterns, histogram of each image is generated. Let $ACbins$ to indicates the total number of bins after merging adjacent patterns. Example of histograms before and after merging adjacent patterns is shown in Figure 3.6.

3.3.4 Construction of DC-Pattern histogram

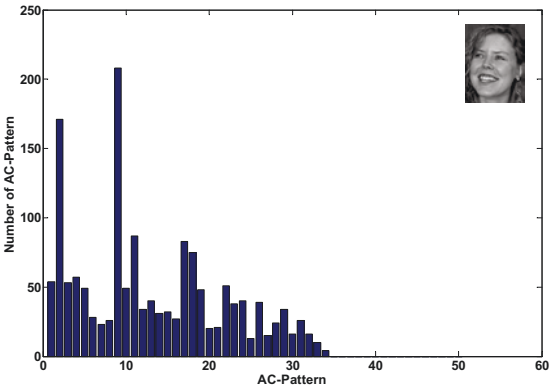
Differently from previous AC-Patterns that describe the local feature information inside each block, DC-Patterns integrate more global features by using gradients between each block and its neighbors. Details can be found in Section 3.2.2. Let $DCbins$ be the number of most frequent DC-Patterns chosen.

3.3.5 Application to face recognition

As our improvements focus on the construction of AC-Patterns, to evaluate these improvements in face recognition, we do two classes of experiments: face recognition by using AC-Patterns alone and by using AC-Patterns and DC-Patterns together. For AC-



(a) Histogram of original AC-Pattern



(b) Histogram of AC-Pattern after merging

Figure 3.6: Histogram of AC-Pattern

Patterns alone, the descriptor is defined as:

$$H = H_{AC} \tag{3.7}$$

For AC-Patterns and DC-Patterns together, the descriptor is defined as the concatenation of H_{AC} and H_{DC}:

$$H = [(1-\alpha) \times H_{AC} \quad \alpha \times H_{DC}] \tag{3.8}$$

where α is a weight parameter that controls the impact of AC-Patterns histogram and DC-Patterns histogram.

To measure the similarity between two descriptors, Manhattan distance is used:

$$Dis(Q, I_i) = \sum_{j=1}^{K} |H_Q(j) - H_{I_i}(j)| \tag{3.9}$$

where K indicates the dimension of the descriptors and H_Q and H_{I_i} are the descriptors of query and i^{th} image in the database.

3.3.6 Performance analysis

To evaluate the performance of the proposed improvements, GTF [80] and ORL [81] databases, two commonly used databases for face recognition, are adopted.

The GTF was created in 1999 at the Center for Signal and Image Processing of Georgia Institute of Technology. It contains 15 images of 50 people. The images show frontal and/or tilted faces with different facial expressions, lighting conditions and scales. The images are at the resolution 640×640 pixels in which the size of face is 150×150 pixels. Each image is manually labeled to determine the position of the face in the image. The information is also used to crop faces manually from images in our experiments.

The ORL was created between 1992 and 1994 at AT&T Laboratories Cambridge. It contains 10 different face images of 40 different persons. For some persons, the images were taken at different times, at varying the lighting, with different facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken with a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

As *Linear scan* was initially verified on these two databases, to demonstrate the contribution of our proposal, we implement *Linear scan* and our improved approach, *Zigzag-Pattern* on these two databases too. Face images cropped from GTF and ORL databases have small sizes of about 120×90 pixels, as shown in Figure 3.7 and Figure 3.8. EER introduced in Section 2.7.4 is used to evaluate the performance.

Results on GTF database

In our experiments, like in [70], we select the first 11 images of each person as training database and remaining 4 images as query images for recognition. Therefore, the total number of images in the training database is 550 (11×50) and that of query images is 200 (4×50).

We first use descriptor of AC-Pattern alone for image retrieval. We should emphasize that for different parameters $QPAC$ (quantization parameter for AC coefficients) and $ACbins$, different performance will be observed. Here we only compare the best performance of each method. Figure 3.9(a) shows the curve of comparison when Nc changes.

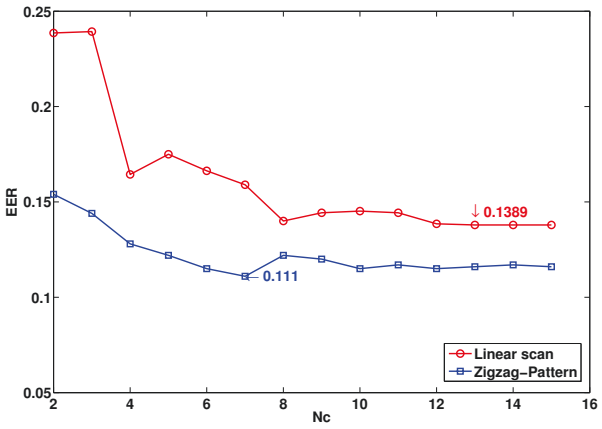
Figure 3.7: 15 different faces of one person in GTF

Figure 3.8: 10 different faces of one person in ORL

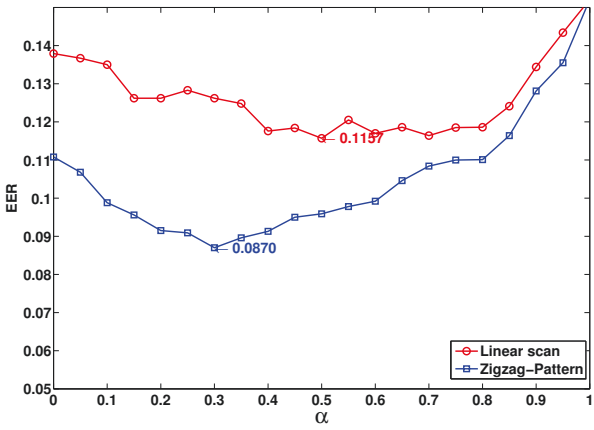
As we can see, *Zigzag-Pattern* outperforms *Linear scan* and details of the comparison are listed in Table 3.1. The *Zigzag-Pattern* can get 20% enhancement comparing with *Linear scan*.

Before using the descriptor of AC-Pattern and DC-Pattern together, the parameter of descriptor of DC-Pattern should be set. After numerous experiments, when $\gamma = 4$, $QPDC = 26$ and $DCbins = 400$, the best performance can be got and the lowest EER is 0.152 when face recognition is only executed by DC-Pattern.

Finally, we use the descriptor of the AC-Patterns and DC-Patterns together to do face recognition. For both methods, we tested different sets of descriptor parameters of AC-Pattern to find the ones that can assure the best performance while the parameters descriptor of DC-Pattern are the same. The weight parameter α is changed to see the global comparison of the performance, as shown in Figure 3.9(b) and the details of comparison are listed in Table 3.1. As they indicate, *Zigzag-Pattern* with DC-Pattern outperforms



(a) EER according to different Nc



(b) EER according to different α

Figure 3.9: Results on GTF

Linear scan with DC-Pattern: 25% improvement can be observed.

Results on ORL database

For tests, like in [70], we use first 6 images of every person as training database and remaining 4 images as query images for recognition. Therefore, the total number of images in the training database is 240 and that of query images is 160.

We do similar experiments as we did on GTF database. Figure 3.10(a) shows the curve of comparison when Nc changes. As we can see, *Zigzag-Pattern* outperforms *Linear scan*: 38% improvement can be got.

The global comparison of the performance of face recognition by concatenation of the AC-Pattern and DC-Pattern histogram is shown in Figure 3.10(b) and the details of

Table 3.1: Comparison between different descriptors on GTF

Descriptor	*Linear scan*	*Zigzag-Pattern*	*Linear scan* +DC-Pattern	*Zigzag-Pattern* +DC-Pattern
Parameters	Nc=13, QPAC=40, ACbins=80	Nc=7, QPAC=10, ACbins=50	Nc=13, QPAC=40, ACbins=80, γ=4, QPDC=26, DCbins=400	Nc=7, QPAC=10, ACbins=50, γ=4, QPDC=26, DCbins=400
EER	0.1379	**0.111**	0.1157	**0.087**

comparison are listed in Table 3.2. As shown in the table, *Zigzag-Pattern* with DC-Pattern outperforms *Linear scan* with DC-Pattern: 17% uplift can be observed.

Table 3.2: Comparison between different descriptors on ORL

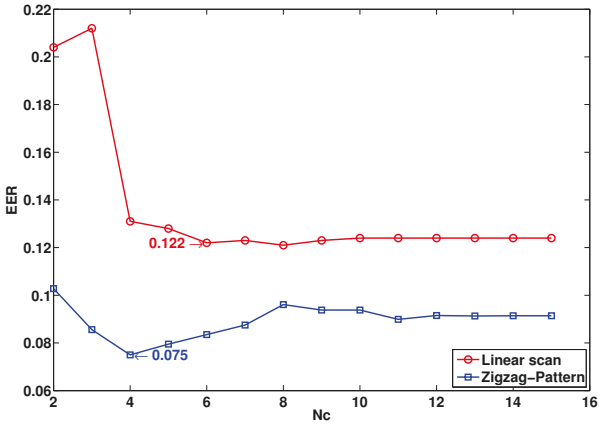
Descriptor	*Linear scan*	*Zigzag-Pattern*	*Linear scan* +DC-Pattern	*Zigzag-Pattern* +DC-Pattern
Parameters	Nc=6, QPAC=30, ACbins=80	Nc=4, QPAC=30, ACbins=100	Nc=15, QPAC=30, ACbins=400, γ=3, QPDC=70, DCbins=250	Nc=4, QPAC=30, ACbins=250, γ=3, QPDC=70, DCbins=250
EER	0.122	**0.075**	0.0607	**0.05**

Conclusion

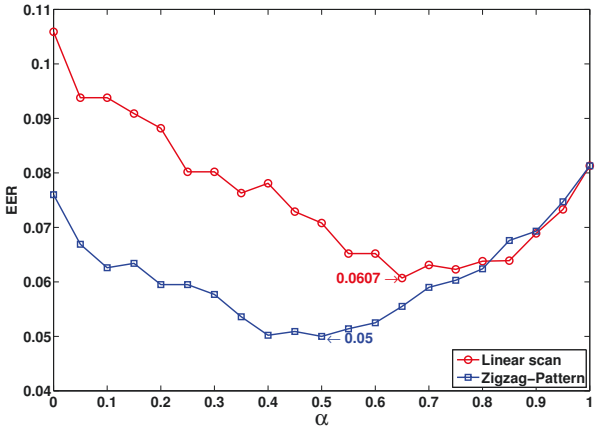
We can conclude, from above experimental results, that our proposal of *Zigzag-Pattern* improves the performance both on GTF and ORL database. Furthermore, fewer AC coefficients and fewer number of bins of histogram are used. This means that the dimension of feature vector and feature descriptor of *Zigzag-Pattern* is smaller than that of *Linear scan*.

3.4 Proposal for face recognition and texture retrieval

In this section, we propose a new and simple but effective approach that can apply both on face database and texture database. A general description is given firstly. And



(a) EER according to different Nc



(b) EER according to different α

Figure 3.10: Results on ORL

then details of approach are described, followed by the experimental results.

3.4.1 General Description

Images are firstly transformed by 4×4 block DCT transform and luminance normalization and quantization presented in Section 3.2.2 are then executed on DCT coefficients. For each block, 9 AC coefficients are selected to construct feature vectors, named *Sum-Pattern*. Finally, the concatenation of Sum-Pattern histogram H_{Sum} and DC-Pattern histogram H_{DC} (see Section 3.2.2) is used for face recognition and texture retrieval.

3.4.2 Sum-Pattern and its histogram

As we know, the wide use of DCT in image compression and image retrieval comes from its capability to compact the energy. It means that much of the energy lies in low frequency coefficients, so that high frequency could be discarded. In other words, only a reduced part of DCT coefficients can efficiently represents the image contents. Furthermore, the DC coefficient indicates the average energy of the block and some AC coefficients contain directional information (See details in Section 2.1).

Inspired from mentioned above, we select 9 AC coefficients in each block to construct Sum-Pattern. 9 coefficients are categorized into 3 directional groups: horizontal, vertical and diagonal. The sums of 2 or 3 coefficients in each group form Sum-Pattern. We use parameter $NcSum$ to indicate the number of coefficients that are used in each group. The process of this construction is shown in Figure 3.11.

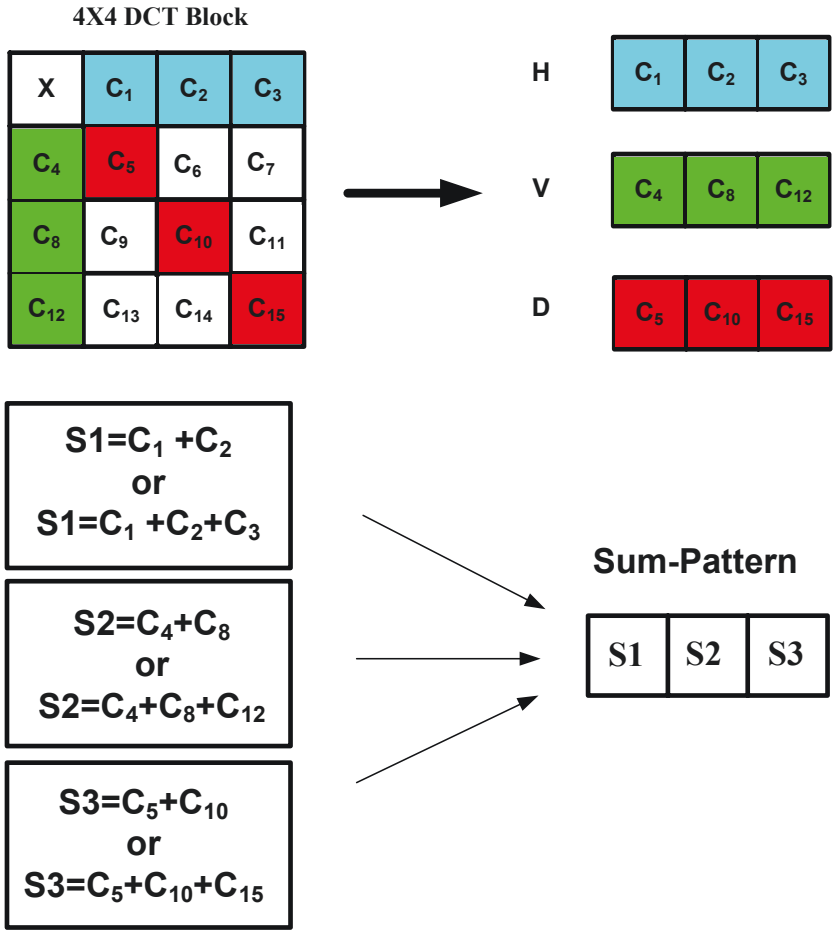


Figure 3.11: Sum-Pattern Construction

Like for previous proposal, we generate the histogram of Sum-Pattern as image descriptor. A disadvantage of the histogram method is that it requires a large number of histogram bins, typically several hundreds, to capture information of feature vector accurately. Thus it leads to complexity in both storage of image features and retrieval timing. To overcome this drawback, in section 3.3, we proposed to merge adjacent patterns. But in this section, we do it in another way: we adopt two improvements.

From the original histogram of Sum-Patterns, shown in Figure 3.12(a), we can make two observations: the first one is that the first Sum-Pattern inside the histogram is very dominant, similar with the one in H_{AC} as described in Section 3.3.3. This Sum-Pattern mainly corresponds to uniform blocks of image and we will not consider this pattern in the Sum-Pattern histogram. The second one is that there is only a part of Sum-Patterns that appears with large number of occurrences and at the opposite, a large number of Sum-Patterns that appears rarely. So in consideration of time-consuming and efficiency, we select the Sum-Patterns which have large number of occurrences to construct the histogram. We use parameter *Sumbins* to represent the number of Sum-Patterns that are selected. For constructing the Sum-Pattern histogram of an image, we calculate the occurrence of these Sum-Patterns in this image, and then we get the Sum-Pattern histogram H_{Sum}, as shown in Figure 3.12(b).

3.4.3 DC-Pattern and its histogram

Same DC-Pattern histogram as presented in Section 3.3.4 is also used in this proposal. Let *DCbins* be the number of DC-Patterns retained. More details of DC-Pattern can be found in Section 3.2.2.

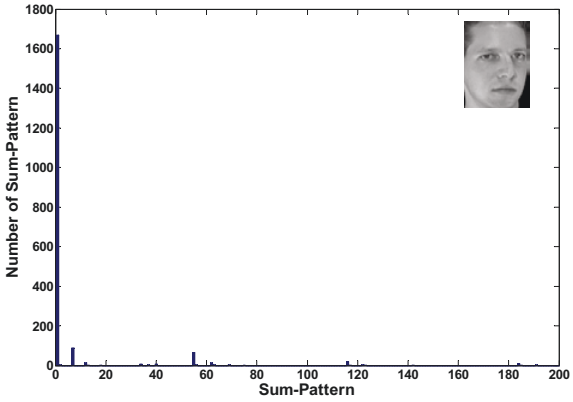
3.4.4 Similarity measurement

We use the concatenation of Sum-Pattern and DC-Pattern histogram as feature descriptor for image retrieval. In this context, the descriptors are defined as follows:

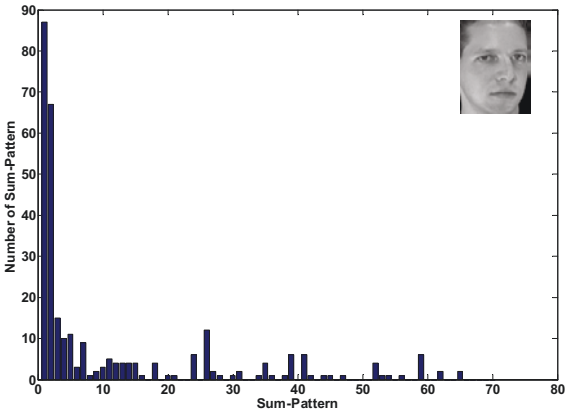
$$H = [(1-\alpha) \times H_{Sum} \quad \alpha \times H_{DC}] \tag{3.10}$$

α is a weight parameter that controls the impact of Sum-Pattern histogram and DC-Pattern histogram.

Beside Manhattan distance, Chi-Squared distance (χ^2 distance) which is widely used



(a) Histogram of original Sum-Patterns



(b) Histogram of selected Sum-Patterns

Figure 3.12: Histogram of Sum-Patterns

for measuring the similarity between histograms is also used as similarity measurement. The definition of Manhattan distance can be found in Equation 3.9 and χ^2 distance is defined as follows:

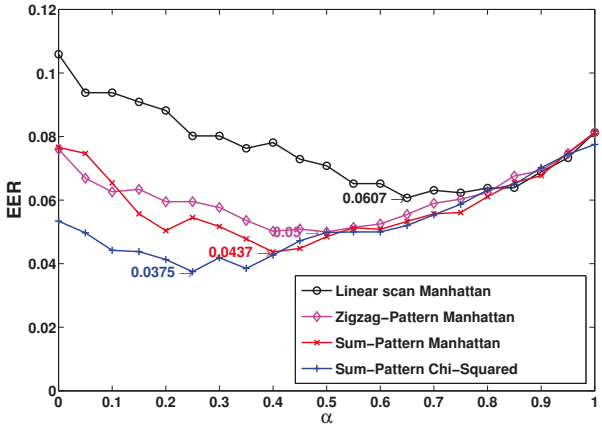
$$Dis(Q, I_i) = \sum_{j=1}^{K} \frac{(H_Q(j) - H_{I_i}(j))^2}{H_Q(j) + H_{I_i}(j)} \tag{3.11}$$

in which H_Q and H_{I_i} are feature descriptors of the query image Q and that of i^{th} image in the database, K indicates the dimension of the descriptors.

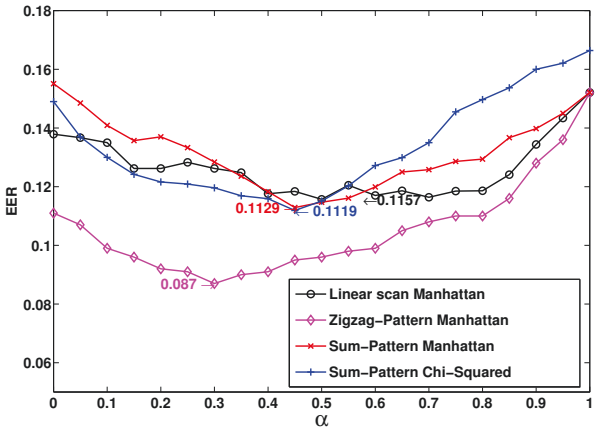
3.4.5 Experimental Results

The objective of the experimental section covers three important issues: first, we evaluate the contribution of Sum-Pattern. Second, we extend the application of proposal from face recognition to texture retrieval. Third, we compare the texture retrieval performance

with previous research work, including *Linear scan*, *Zigzag-Pattern* and state-of-the-art methods in wavelet domain presented in [82–84].



(a) Performance on ORL



(b) Performance on GTF

Figure 3.13: Global comparison of different methods

Face Recognition

Same experiments on GTF and ORL database described in Section 3.3 are executed to demonstrate the contribution of *Sum-Pattern*. The descriptors of feature vectors constructed from AC coefficients are firstly used alone for face recognition. For *Sum-Pattern* method, we tested different distances: Manhattan distance and Chi-Squared distance. And also, different $NcSum$ are tested. Table 3.3 gives the details of comparison. From this table, we can conclude that *Sum-Pattern* with χ^2 distance outperforms on ORL database and *Zigzag-Pattern* outperforms on GTF database. When $NcSum = 2$, EERs are lower

than that of when $NcSum = 3$ no matter Manhattan distance or Chi-Squared distance is used in ORL database, while EERs are same level in GTF database. Furthermore, χ^2 distance outperforms Manhattan distance in this task: it can always assure lower EERs.

Table 3.3: Comparison of different feature descriptors of AC coefficients

Methods	ORL		GTF	
	Parameters	EER	Parameters	EER
Linear scan (Manhattan distance)	Nc=6, QPAC=30, ACbins=80	0.122	Nc=13, QPAC=40, ACbins=80	0.1389
Zigzag-Pattern (Manhattan distance)	Nc=4, QPAC=30, ACbins=100	0.075	Nc=7, QPAC=10, ACbins=50	**0.111**
Sum-Pattern (Manhattan distance)	NcSum=2, QPAC=30, Sumbins=110	0.0766	NcSum=2, QPAC=10, Sumbins=160	0.1515
Sum-Pattern (Chi-Squared distance)	NcSum=2, QPAC=30, Sumbins=50	**0.0515**	NcSum=2, QPAC=30, Sumbins=40	0.1490
Sum-Pattern (Manhattan distance)	NcSum=3, QPAC=30, Sumbins=80	0.0813	NcSum=3, QPAC=10, Sumbins=150	0.1530
Sum-Pattern (Chi-Squared distance)	NcSum=3, QPAC=30, Sumbins=50	0.0560	NcSum=3, QPAC=30, Sumbins=70	0.1447

And then, the concatenation of Sum-Pattern histogram and DC-Pattern histogram is evaluated. We change the weight parameter α to see the global comparison of the performance, as shown in Figure 3.13. Table 3.4 gives the details of the comparison. From these comparisons, we can see that, *Sum-Pattern* combined with DC-Pattern outperforms AC-Pattern combined with DC-Pattern on ORL database: improvement increase by 38%. On GTF database, it gets the similar performance as *Linear scan* combined with DC-Pattern but has a lower performance than *Zigzag-Pattern* combined with DC-Pattern. However, the dimension of Sum-Pattern is 3, while that of AC-Pattern in *Linear scan* is much larger than 3: 15 maximal.

Texture retrieval

As we want to extend our proposal to a wider application field, we also evaluate our proposal in the context of texture retrieval. Vision Texture database (VisTex) [85] is chosen for evaluation. The whole VisTex texture database has 167 natural textured

Table 3.4: Comparison of EER on ORL and GTF

Methods	ORL		GTF	
	Parameters	EER	Parameters	EER
Linear scan +DC-Pattern (Manhattan distance)	Nc=6, QPAC=30, ACbins=80	0.0607	Nc=13, QPAC=40, ACbins=80	0.1157
Zigzag-Pattern +DC-Pattern (Manhattan distance)	Nc=4, QPAC=30, ACbins=100	0.05	Nc=7, QPAC=10, ACbins=50	**0.087**
Sum-Pattern +DC-Pattern (Manhattan distance)	NcSum=2, QPAC=30, Sumbins=110	0.0437	NcSum=2, QPAC=10, Sumbins=140	0.1129
Sum-Pattern +DC-Pattern (Manhattan distance)	NcSum=3, QPAC=30, Sumbins=80	0.0523	NcSum=3, QPAC=10, Sumbins=140	0.1130
Sum-Pattern +DC-Pattern (Chi-Squared distance)	NcSum=2, QPAC=30, Sumbins=70	**0.0375**	NcSum=2, QPAC=30, Sumbins=40	0.1119
Sum-Pattern +DC-Pattern (Chi-Squared distance)	NcSum=3, QPAC=30, Sumbins=80	0.382	NcSum=3, QPAC=10, Sumbins=140	0.1145

images. To compare with the other approaches, we evaluate our proposal on a classical selection of 40 textures which have already been extensively used in texture image retrieval literature [82–84]. We named this selection as Small VisTex. This selection is displayed in Figure 3.14.

In the experiments, the 512 × 512 color version of the textures are divided into 16 non-overlapping subimages (128 × 128) and converted to gray scale images, thus creating a database of 640 images belonging to 40 texture-classes, each class includes 16 different samples. In the process of retrieval, each image is used once as query image. The relevant images for each query consist of all the subimages from the same original texture. Like in other literatures, we use the average retrieval rate (ARR) to evaluate the performance. For comparison purpose, we retrieve 16 images for each query. Every subimage in the database is used as query once for retrieval and finally ARR is calculated.

Table 3.5 provides a detailed comparison of ARR for 3 wavelet-based approaches re-

Figure 3.14: Selected textures from VisTex database

ported in [82–84] belonging to the state-of-the-art and Sum-Patter with DC-Pattern, where the highest ARR is marked bold. In [82], images are decomposed by Discrete Wavelet Transform (DWT), coefficients in each subband are modeled by a Generalized Gaussian Density (GGD). Similarity is measured by computing KL-distances between model parameters. In [83], images are decomposed by Rotated Complex Wavelet Filters (RCWF) and DT-CWT, and feature vectors are formed by the energy and standard deviation of each subband; similarity is measured by Canberra distance. In [84], images are decomposed by Dual-Tree Complex Wavelet Transform (DT-CWT), and each detail subband coefficients are modeled by Gamma distribution; similarity is measured by KL-distance. It can be observed that the combination of Sum-Pattern and DC-Pattern adopting either Manhattan distance or χ^2 distance as similarity measurement outperforms referred methods .

Table 3.5: Comparison of ARR on Small VisTex

Method	DWT [82]	DT-CWT [84]	RCWF [83]	Sum-Pattern +DC-Pattern	
				Manhattan	Chi-Squared
ARR(%)	76.30	81.73	82.34	83.78	**84.71**

Conclusion on experiments

From above experiments as well on face database as on texture database, we can get two conclusions: first, *Sum-Pattern* improves the performance of face recognition on ORL database, and could get similar performance on GTF database, but with the advantage of smaller dimension of feature vector. Second, Sum-Pattern with DC-Pattern works well in the application of texture retrieval, especially adopting χ^2 distance as similarity measurement. It outperforms state-of-the-art methods in wavelet domain.

3.5 Proposal for color texture retrieval

Based on the previous works described in Section 3.3 and Section 3.4, a new approach using the combination of texture feature and color feature for color texture image retrieval is proposed in this section.

3.5.1 General Descriptions

To take color into consideration, we make the choice of YCbCr color space which is classically adopted in JPEG standard. In this color space, there are three components: one is luminance component I_Y, and the other two are chroma components I_{Cb} and I_{Cr}. Then each component is decomposed into 4x4 blocks which are transformed by DCT. So for each DCT block we get 1 DC coefficient and 15 AC coefficients . Furthermore, same preprocessing steps as described in section 3.3.2 are applied to the DCT coefficients of luminance component I_Y to eliminate the effect of luminance variation. And then the DCT coefficients are quantized with a quantization coefficient QP after normalization. D_Y, D_{Cb} and D_{Cr} represent the DCT coefficients after normalization and quantization from each component. As the I_Y component can be seen as a gray-level copy of the original color image, and as the texture feature is considered as mainly appearing in the luminance component of the image, then the texture feature is extracted from this component: 9 AC coefficients of every block in D_Y are selected to construct Texture-Pattern. As the DC coefficient in each block reflects the average value of each block, Color-Pattern is constructed by DC coefficients from each component, D_Y, D_{Cb} and D_{Cr}. Finally, we use the histogram of Texture-Pattern H_T and histogram of Color-Pattern H_C as feature descriptors for image retrieval. The block diagram of the proposed approach is shown in Figure 3.15.

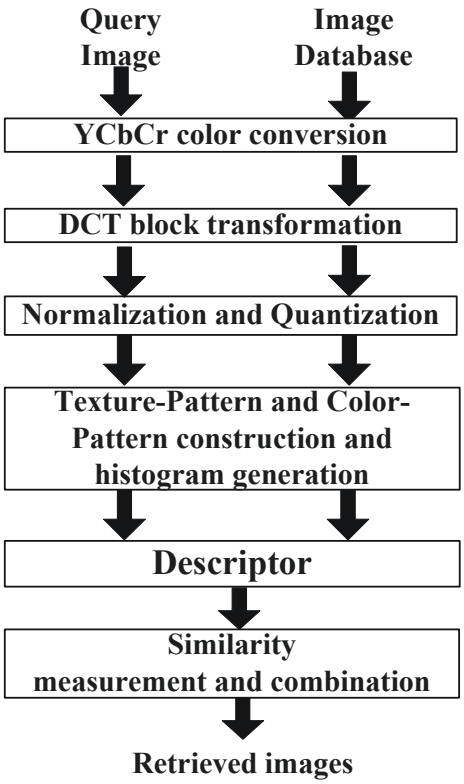


Figure 3.15: Block diagram of proposal

3.5.2 Texture-Pattern construction

Texture-Pattern is obtained from the AC coefficients of D_Y. 9 coefficients out of all 15 AC coefficients are selected in each block and categorized into 3 groups: horizontal (Group H), vertical (Group V) and diagonal (Group D), according to the directional information they represented, as shown in Figure 3.16. For each group, the sum of the coefficients is calculated firstly and then the squared-differences between each coefficient and the sum of this group are calculated. Finally, the sums of these squared-differences of each group are used to construct Texture-Patterns.

3.5.3 Color-Pattern construction

The DC coefficient reflects the average value of each block, DC coefficients in D_Y can represent the average luminance of each DCT block and DC coefficients in D_{Cb} and D_{Cr} can be seen as the average chrominance of each block: these three together can represent luminance and chroma information of each block. We will call this as color information.

From the above observation, the Color-Pattern is constructed by the DC coefficients

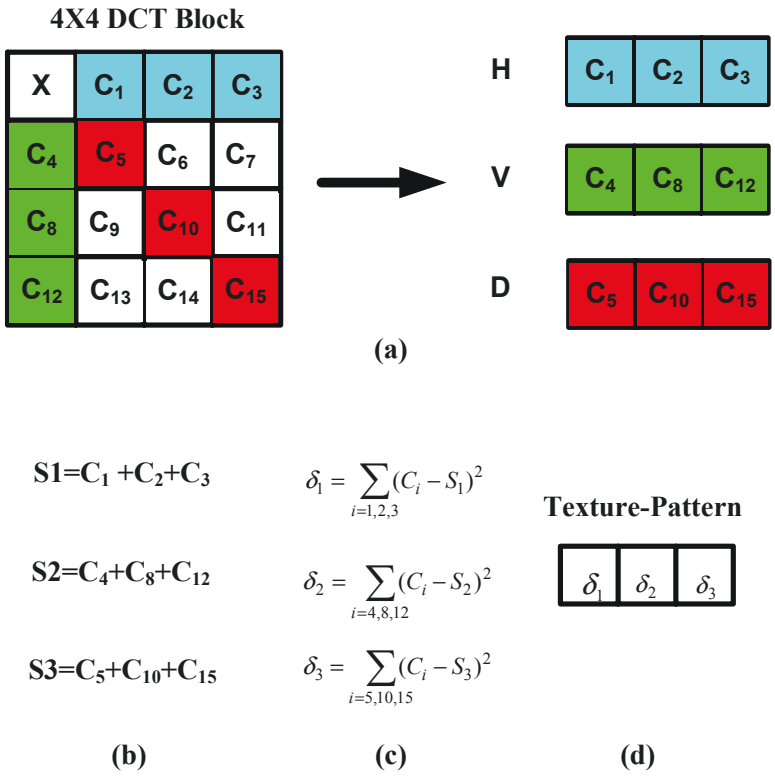


Figure 3.16: Texture-Pattern:
(a) Three groups of AC coefficients extracted from DCT block (b) Sums of each group (c) Sums of squared-differences (d) Texture-Pattern

from the 3 components of each block in the image. The procedure of forming Color-Pattern is shown in Figure 3.17.

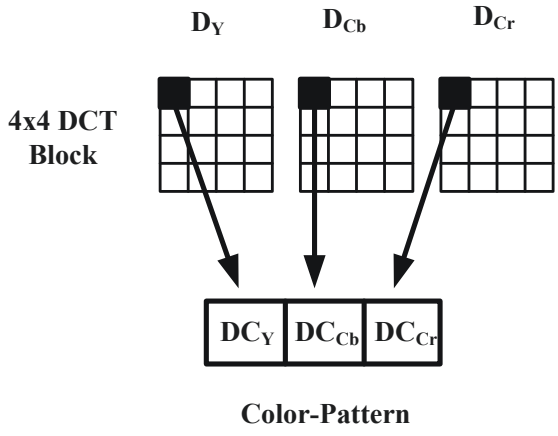
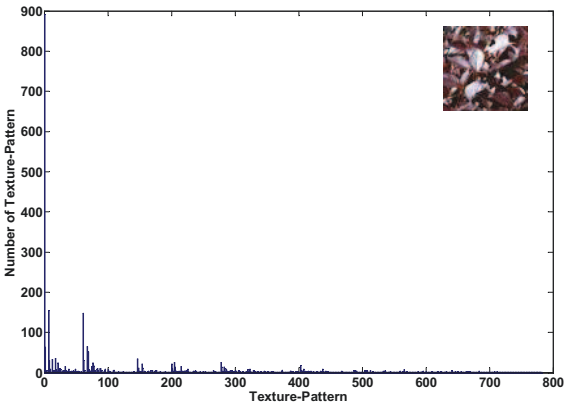


Figure 3.17: Color-Pattern

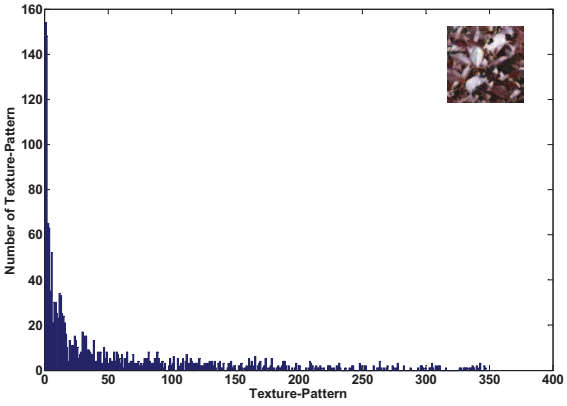
3.5.4 Histogram generation

The original histogram of Texture-Pattern is shown in Figure 3.18(a). We will have the same considerations on this histogram as those we had in Section 3.4.2: we select those of Texture-Patterns which have higher frequency of occurrence to construct the descriptor H_T. We will use parameter $Tbins$ to represent the number of bins that are selected in the histogram of Texture-Patterns; the first Texture-Pattern inside the histogram that corresponds to uniform blocks of image will not be considered as a representative pattern in the Texture-Pattern histogram.

So the histogram of Texture-Patterns that will be used to image retrieval is as shown in Figure 3.18(b). This histogram is obtained by selecting the first 350 ($Tbins = 350$) highest frequencies of occurrence Texture-Patterns from the histogram of Figure 3.18(a).



(a) Histogram of original Texture-Pattern



(b) Histogram of selected Texture-Pattern

Figure 3.18: Histogram of Texture-Patterns

Finally, we get H_T from Texture-Pattern and H_C from Color-Pattern as the feature descriptors which are used for image retrieval.

3.5.5 Similarity measurement

As we stated in Section 3.4, χ^2 distance (see Equation 3.11) is more suitable for measure the similarity between histograms, we choose χ^2 distance as the similarity measurement in this approach.

Since we have two kinds of feature descriptors, the texture descriptor H_T and the color descriptor H_C, we will have two sets of distances that will be fused to measure the similarity of images. However, each feature descriptor has its own physical meaning, and their values range differently, so before fusing their distances, they should be normalized. This can be done through the following ways: given a query image, by calculating distances of texture descriptors and that of color descriptors between this query and all images in the database. Thus two sets of distances $\{Dis_T(i)\}$ and $\{Dis_C(i)\}$ are obtained, where $i = 1, \cdots, N$. N is the number of images in the database. $Dis_T(i)$ is the distance between texture descriptor of query image Q and i^{th} image in the database, and $Dis_C(i)$ is the distance between color descriptors of query image Q and i^{th} image in the database. Thus the distance normalization can be implemented as:

$$\begin{aligned} Dis_T^N(Q, I_i) &= \frac{Dis_T(i) - min\{Dis_T(i)\}}{max\{Dis_T(i)\} - min\{Dis_T(i)\}} \\ Dis_C^N(Q, I_i) &= \frac{Dis_C(i) - min\{Dis_C(i)\}}{max\{Dis_C(i)\} - min\{Dis_C(i)\}} \end{aligned} \tag{3.12}$$

where $Dis_T^N(Q, I_i)$ and $Dis_C^N(Q, I_i)$ are the normalized distances between texture and color descriptors of query image Q and i^{th} images in the database respectively. Both type of distances now range from 0 to 1.

The global distance that is used to evaluate the similarity between the query and images in the database is then given by:

$$Dis_G(Q, I_i) = (1 - \beta) \times Dis_T^N(Q, I_i) + \beta \times Dis_C^N(Q, I_i) \tag{3.13}$$

where $\beta \in \{0, 1\}$ is a weight parameter that can control the impact of color feature and texture feature in the procedure of image retrieval.

3.5.6 Experimental results

As Texture-Pattern is also constructed from AC coefficients of each DCT block, the experiments are divided into two main parts: the first one is to evaluate the contribution

of Texture-Pattern. So we execute face recognition on three face databases and texture retrieval on one texture database by *Texture-Pattern* and compare with the results of *Linear scan*, *Zigzag-Pattern* and *Sum-Pattern*. The second one is to evaluate the performance of color texture retrieval by combination of proposed texture feature and color feature. Experiments are implemented on two data sets of texture images and the results are compared with previous works and state-of-the-art methods.

Evaluation of Texture-Pattern

In this part, we perform experiments on face databases firstly. As ORL and GTF are relatively small database, to further evaluate our proposal, we also implement it on FERET database [86].

The FERET [86] database is a large collection of facial images. This database was created between 1993 and 1996 at George Mason University, which contains 1564 groups of images for a total of 14126 images that includes 1199 individuals and 365 duplicate groups of images. A duplicate group is a second set of images of a person already in the database and was usually taken on a different day. These images are divided into several sets. Here two sets of frontal view faces **fa** and **fb** were selected to evaluate the proposed method: **fb** is used as query images for retrieval from the **fa**. The position data of the eyes, nose and mouth for each image are also provided with the database. Depending on whether this information is used, the experiments can be divided into two types: fully automatic and partially automatic [86]. Our experiments are of the second type, that means, faces are extracted from the images based on the position data. And then, faces are cropped to extract the region of interest to remove background and hairs. Finally, they are scaled to 150×130 pixels. Figure 3.19 shows the examples of original images and faces used in the experiments.

Three different methods in previous sections are compared with Texture-Pattern: *Linear scan*, *Zigzag-Pattern*, and *Sum-Pattern*. The comparison of experimental results is shown in Table 3.6. As described before, different parameters could lead to different performance, only the best results are listed in the table. From this table, we can see that, *Texture-Pattern* outperforms referred methods both on ORL and FERET database. But on GTF database, it ranks 2^{nd} after *Zigzag-Pattern*.

Texture retrieval is then executed on a classical selection of 40 textures from VisTex [85] as described in Section 3.4 named Small VisTex. Table 3.7 provides the comparison of

(a) Original images on FERET

(b) Cropped Faces

Figure 3.19: FERET database

ARR on VisTex database and Figure 3.20 provides the comparison in the form of Precision-Recall pair. Similar with the comparison on faces database, the four different methods are compared. From this table and this figure, we can conclude that: *Texture-Pattern* still outperforms in the experiments of texture retrieval by feature descriptors constructed from AC coefficients.

Furthermore, if the concatenation of feature descriptors constructed from AC coefficients (AC-Pattern histogram, Zigzag-Pattern histogram, Sum-Pattern histogram and

Table 3.6: EER obtained for different feature vectors from AC coefficients

Methods of constructing features from AC coefficients	ORL	GTF	FERET
Linear scan	0.0949	0.1389	0.0511
Zigzag-Pattern	0.0750	**0.111**	0.0546
Sum-Pattern	0.515	0.1447	**0.0496**
Texture-Pattern	**0.0479**	0.1219	**0.0496**

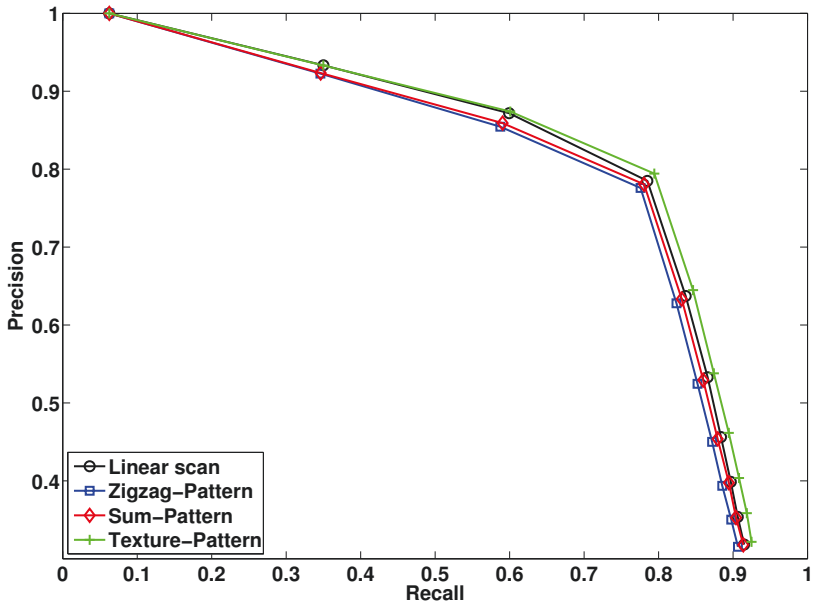


Figure 3.20: Comparison of different feature vectors of AC coefficients on Small VisTex

Table 3.7: ARR obtained on Small VisTex (Gray texture)

Method	*Linear scan*	*Zigzag-Pattern*	*Sum-Pattern*	*Texture-Pattern*
ARR(%)	73.90	77.01	78.06	**79.43**

Texture-Pattern histogram) and DC-Pattern histogram described in Section 3.3.4 is used to do texture retrieval, better performance can be observed. Table 3.8 and Figure 3.21 provide the details of comparison from the point of view of ARR and Precision-Recall pairs. In the table, "DT-CWT+RCWF" indicates the method proposed in [83], in which dual-tree complex wavelet transform (DT-CWT) and dual-tree rotated complex wavelet filters (DT-RCWF) are used to decompose the images and the energies and standard deviations of each subband are used as the feature descriptors. "DT-CWT" represents the method presented in [84], in which Dual-tree complex wavelet transform (DT-CWT) is used to decompose images and Gaussian Mixture Models (GMM) are used to model the magnitudes of detail subband coefficients. From the table, it can be observed that Texture-Pattern with DC-Pattern outperforms other methods, including state-of-the-art methods in wavelet domain [83, 84].

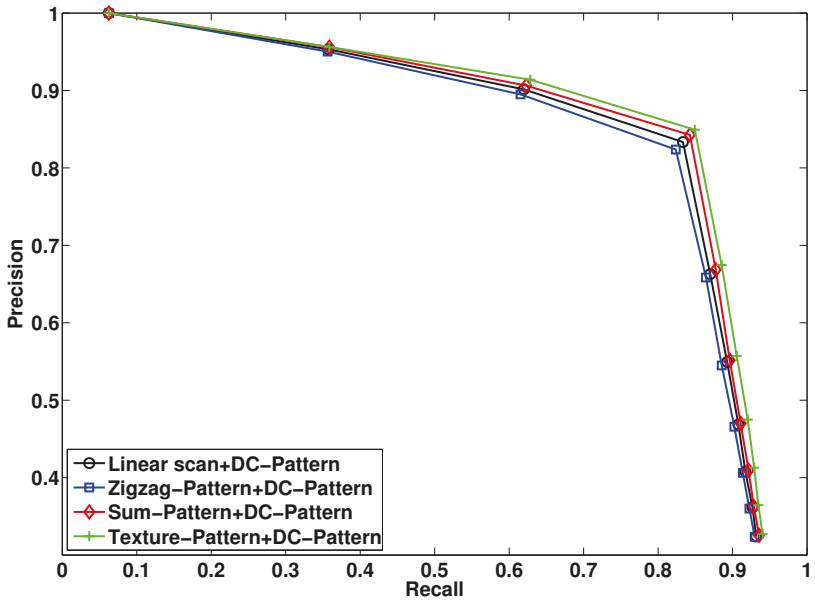


Figure 3.21: Comparison of Precision and Recall on Small VisTex

Table 3.8: Comparison of ARR on Small VisTex (Gray texture)

Method	DT-CWT	DT-CWT +RCWF	*Linear scan*	*Zigzag Pattern*	*Sum-Pattern*	*Texture-Pattern*
ARR(%)	81.73	82.34	82.09	82.59	84.71	**85.20**

Color texture retrieval by using both color and texture features

We perform experiments of image retrieval on color VisTex texture database. To compare with the other approaches, we evaluate our proposal on two sets of VisTex: one is Small VisTex used in our previous experiments of texture retrieval. And the other is the whole collection of VisTex, that means the selection of 167 classes of texture.

We also should emphasize that for different β in Equation 3.13, various ARR can be got because of different impact of color and texture feature in the process of retrieval. All the results presented below are the ARR when $\beta = 0.35$. This value is found experimentally to assure the best ARR that we can get.

Table 3.9 presents the comparative experimental results on the data set of 40 texture classes with referred methods. In [87], images are decomposed by complex wavelet transform, coefficients in each subband are modeled by Gaussian Copula with Gamma (GCG) and statistics and marginal parameters of each band form feature vectors. In [88], coefficients in each subband are modeled by Student-t distribution and Rao geodesic distance

is used to measure the similarity. These two approaches are considered as state-of-the art approaches. The comparison shows that our proposal performs better.

Table 3.9: ARR on Small VisTex (color version)

Method	GCG [87]	Student-t [88]	Texture-Pattern+Color-Pattern
ARR(%)	85.83	89.65	**90.16**

Table 3.10 presents the retrieval performance on the whole VisTex database. In [89], Gabor filters are used to extract features from R,G,B components, and in [90], authors proposed Chromatic Statistical Landscape Features (CSLF) to represent the color texture. From this table, we can see that as many classes of texture in VisTex are not homogeneous, the retrieval rate is much lower than that of 40 classes. But our proposal still outperforms referred methods.

Table 3.10: ARR on the whole VisTex (color version)

Method	Gabor [89]	CSLF [90]	Texture-Pattern+Color-Pattern
ARR(%)	52.0	56.2	**58.09**

3.6 Conclusion

In this chapter, we have made a comprehensive study of image retrieval based on the histogram of patterns constructed from coefficients of block DCT.

First, we introduced the general principle of image retrieval using the histogram of patterns constructed from DCT blocks after detailing related works.

Secondly, an improved method *Zigzag-Pattern* was proposed. Two aspects of improvement have been presented: 1) zigzag is adopted as the way of arranging coefficients in AC-Pattern; 2) adjacent patterns are defined and merged to reduce the number of bins of the histograms used for retrieval. Experimental results show that these two improvements can enhance the performance in ORL and GTF databases compared with the original approach.

Then with the consideration of the capability of DCT for compacting energy, a new proposal for face recognition and texture retrieval is presented. This proposal has two contributions: 1) Sum-Pattern, a new kind of feature vector constructed from AC coefficients,

is proposed. With this feature vector, only a 3-D vector could represent the information of AC coefficients efficiently for face recognition or texture retrieval. 2) a selection of most frequent patterns is used to construct the histograms which are used for face recognition or texture retrieval. This process could reduce the dimension of feature descriptors. And we have also evaluated the extensibility of our proposal by applying it on two different kinds of database: face database, which has structural contents, and texture database, which has both structural and unstructured contents.

Finally, a new approach for color texture retrieval is proposed. This proposal has two contributions: 1) it uses the statistical information of coefficients which represent directional features to construct Texture-Pattern. Compared with AC-Pattern, Zigzag-Pattern and Sum-Pattern, this feature vector is more powerful both on face recognition and texture retrieval. 2) it proposes the method of constructing Color-Pattern. Experimental results show that color texture retrieval with combination of Texture-Pattern and Color-Pattern could get better performance than referred methods, including several state-of-the-art approaches in wavelet domain.

In the next chapter, we will change the core of our tools and two methods of color texture retrieval in wavelet domain will be presented.

4 Image descriptors in Wavelet domain

4.1 Introduction

Discrete Wavelet Transform (DWT) is also a powerful tool to extract features from images. So in this chapter, two approaches for color texture image retrieval in wavelet domain are proposed. For retrieval efficiency, the combination of texture feature and color feature is used. As we said in Section 3.2, there are two categories of extracting color and texture features: separately and jointly [29]. Separately means that color images are transformed into luminance and chrominance components and then color and texture feature are extracted separately. Our first proposed method can be classified as this kind. Our second method is in the context of jointly, which means that features are jointly extracted from different spectral bands of color image.

This chapter is organized as follows: related works in wavelet domain are introduced firstly, followed by the proposal for color texture retrieval based on data clustering (K-means) in Section 4.4. Then a sparse representation based approach is presented in Section 4.5. Experimental results of these two approaches are analyzed in Section 4.6.

4.2 Related works

Wavelet has been widely applied for texture retrieval. A popular concept is the framework of probabilistic image retrieval. To the best of our knowledge, this idea is first introduced by Vasconcelos and Lippman in [91–93], which is proposed initially in DCT domain, followed by Do and Vetterli in [82] and Kwitt and Uhl in [84] who extended this framework to wavelet domain. In the probabilistic framework, each image is represented by statistical model and image similarity is measured by a function of these models. For

example, in [82], DWT is used to decompose images and the detail subband coefficients is modeled by Generalized Gaussian Distributions (GGD). A closed-form solution of the KL-divergence between GGDs is used to measure the similarity between two images. In [84], Dual-tree complex wavelet transform (DT-CWT) is used to decompose images and then Rayleigh, Weibull, Gamma and Gaussian Mixture Models (GMM) are used to model the magnitudes of detail subband coefficients. As for similarity measure, KL-divergence is used again. All afore-mentioned approaches are in the context of gray-level texture retrieval; color texture retrieval in the probabilistic framework has also appeared these years. In [94], wavelet coefficients are modeled by Multivariate Generalized Gaussian (MGG) jointly in each color components of texture and geodesic distance is used for similarity measure. In [87], images are decomposed by complex wavelet transform, coefficients in each subband are modeled by Gaussian Copula with Gamma (GCG) and statistics and marginal parameters of each band form feature vectors. A state-of-the-art method of color texture retrieval is presented in [88], in which Laplace and student-t distribution are used to model the color cue and spatial dependencies of wavelet coefficients and geodesic distance is used again for similarity measure. Another state-of-the-art approach is presented in [95], in which wavelet coefficients are modeled by several distributions, but Gaussian Copula with Weibull distribution (GCWD) outperforms. These four approaches will be used as references for our retrieval experiments.

Different with the framework of probabilistic image retrieval, image can be represented by different kinds of features that are extracted from wavelet coefficients by signal processing techniques and by calculating the distance between features, similarity can be got. To the best of our knowledge, the first try in this context in wavelet domain is presented by Chen in [96]. In that approach, "unichrome" features and "opponent" features computed from DWT coefficients are jointly used for image retrieval. In [97], Tian and Mei proposed the circular region energy of coefficients in low frequency bands as color feature and synthesize energy of coefficients in high frequency bands as texture feature. Linear combination of these two feature is applied for image retrieval. In [98], Liapis and Tziritas presented a method in which color feature is represented by 2-D histogram of *CIE Lab* chromaticity coordinates and texture features are extracted using Discrete Wavelet Frames (DWF) analysis. In [99], Young and etc. proposed using the autocorrelogram of wavelets coefficients extracted from Hue and Saturation components as color feature, and using the first and second moments of the BDIP (block difference of inverse probabilities) and BVLC

(block variation of local correlation coefficients) for each subband of Value component as texture feature.

Most of the presented works didn't consider the situation of retrieving compressed format images, especially JPEG 2000 images. For example, if two state-of-the-art methods are applied on JPEG 2000 images, images should be de-compressed entirely firstly, and then transformed by dual-tree complex wavelet transform: this is time consuming! So we propose to extract features directly from DWT. If our proposals are used to retrieve JPEG 2000 images, only partial-decompression is needed: it means that the compressed images are only decoded into wavelet coefficients and then these coefficients are used for retrieval.

Furthermore, most of approaches using the wavelet coefficients to construct feature vectors often focus on the features of subband of wavelet: mean value, energy, stand deviation of each subband are often used as feature vector or the distribution of coefficients in each subband are represented by statistical models. In the following proposals, we will present simple ways to construct feature vectors directly from the coefficients.

4.3 Wavelet decomposition

Color images are firstly converted to YCbCr, whose components are I_Y, I_{Cb} and I_{Cr}. And then, each component is decomposed by N-level Discrete Wavelet Transform (DWT). Results are represented by W_S^{mn}, where $S \in \{Y, Cb, Cr\}$ denotes the components and $m \in \{LL, HL, LH, HH\}$ denotes the subband orientation and $n = \{1, 2, \ldots, N\}$ the wavelet decomposition level. In our approach, we choose CDF 9/7 wavelets that is also used in JPEG2000. Subbands can be classified into two categories: approximation subband W_S^{LLN} and detail subbands W_S^{HLn}, W_S^{LHn} and W_S^{HHn}, as shown in Figure 4.1. In this figure, we demonstrate the decomposition level $N = 2$ as example.

4.4 Descriptor of color texture generated by K-means

This section details our first new approach using the combination of texture and color features for color texture image retrieval in wavelet domain based on data cluster (K-means).

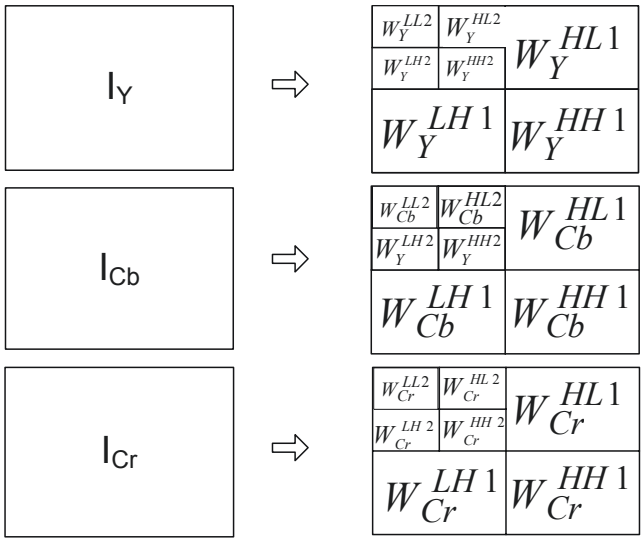


Figure 4.1: Wavelet decomposition

4.4.1 Multiresolution texture-vectors and color-vector

As the I_Y component can be seen as a gray-level copy of the original color image, and as the texture feature is considered as mainly appearing in the luminance component of the image, the multiresolution texture-vectors are constructed from wavelet coefficients in detail subbands of Y channel in both decomposition levels, that's to say W_Y^{HLn}, W_Y^{LHn} and W_Y^{HHn}, where $n = \{1, 2, \ldots, N\}$, as marked blue in Figure 4.2. On the other hand, color-vector is constructed from the coefficients of approximation subbands of each component, that's to say W_Y^{LLN}, W_{Cb}^{LLN} and W_{Cr}^{LLN}, as marked yellow in Figure 4.2.

We will have N kinds of texture-vectors when images are decomposed by N levels. They are constructed by the coefficients at the same position from detail subbands of each decomposition level in Y component. We use $N = 2$ as examples again. In this situation, two levels of resolution can be got, respectively high and low and the texture-vectors are referred as: Hi Resolution texture-vector (TV_1) and Low Resolution texture-vector (TV_2). TV_1 contains three coefficients from three subbands W_Y^{HL1},W_Y^{LH1} and W_Y^{HH1} respectively in first decomposition level. TV_2 is constructed in the same way, but in second decomposition level from three different subbands W_Y^{HL2},W_Y^{LH2} and W_Y^{HH2} respectively. Thus each texture-vector contains vertical, horizontal and diagonal information. Color-vector (CV) is built in the similar way. Each color-vector includes three coefficients at the same position from the lowest-frequency subbands of Y, Cb and Cr components, W_Y^{LL2}, W_{Cb}^{LL2} and W_{Cr}^{LL2} respectively. Figure 4.2 graphically shows the process and the definition

of texture-vectors and color-vector are listed as follows:

$$\begin{aligned} CV &= [W_Y^{LLN}(x_C, y_C),\ W_{Cb}^{LLN}(x_C, y_C),\ W_{Cr}^{LLN}(x_C, y_C)] \\ TV_n &= [W_Y^{HLn}(x_n, y_n),\ W_Y^{LHn}(x_n, y_n),\ W_Y^{HHn}(x_n, y_n)] \end{aligned} \tag{4.1}$$

where (x_n, y_n), $n = \{1, 2, \ldots, N\}$ and (x_C, y_C) indicate the coordinates of the coefficients in each subband. These 3-dimension vectors are used to construct feature descriptors. By this way, only 50% wavelet coefficients are used for constructing feature vector when $N = 1$ and this number decreases to 37.5% when $N = 2$, and 33.4% when $N = 5$.

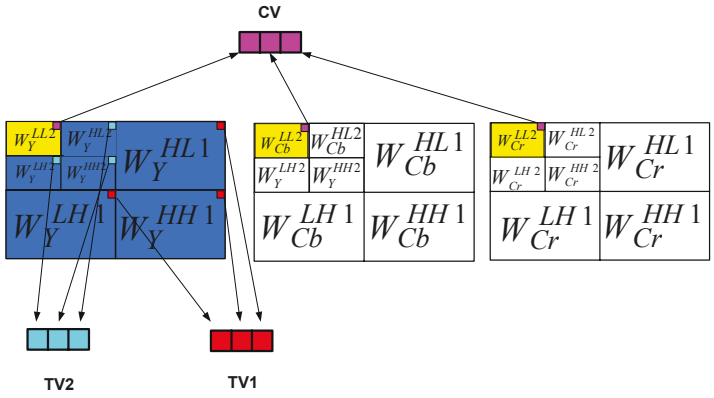


Figure 4.2: Mapping coefficients into vectors (N=2)

4.4.2 Descriptor construction

We use the histogram of these vectors as the descriptor of the image. With the objective of reducing dimensions of descriptors, before generating the histogram, K-means algorithm [100] is used to divide the color-vector space and texture-vector space into partitions that are represented by the cluster centers issued from K-means. And the histogram of vectors is defined as the number of vectors that fall into these partitions. As we detailed in Section 2.4.1, determining a meaningful and appropriate K by objective criterions is still an unsolved problem; the number of clusters K for different kinds of vectors that assure best retrieval performance are found experimentally: for texture-vectors they are all clustered into 400 centers respectively ($K_1 = 400, K_2 = 400, \ldots, K_N = 400$), and for color-vectors, $K_C = 2000$. Therefore each histogram of texture-vectors H_{TVn} has 400 bins respectively. The histogram of color-vectors H_{CV} has 2000 bins.

4.4.3 Similarity measurement

As mentioned in previous chapter, χ^2 distance is more suitable for measuring the similarity between histograms, so in this approach, it is also chosen. The definition of this distance is reminded as follows:

$$Dis(Q, I_i) = \sum_{j=1}^{K} \frac{(H_Q(j) - H_{I_i}(j))^2}{H_Q(j) + H_{I_i}(j)} \tag{4.2}$$

in which H_Q and H_{I_i} are feature descriptors of the query image Q and that of i^{th} image in the database, K indicates the dimension of the descriptors.

Since we have two kinds of feature descriptors: texture descriptor H_{TV} and color descriptor H_{CV}, we will have two sets of distances, and the fusion of both sets of distances is used to determine the similarity of images. However, each feature descriptor has its own physical meanings, and its ranges of values are totally different, so before using the fusing distances of different descriptors, they should be normalized.

Distances are normalized through the ways described in Section 3.5.5.

Let $Dis^N_{TV}(Q, I_i)$ and $Dis^N_{CV}(Q, I_i)$ be the normalized distances of texture and color descriptors respectively, in which:

$$Dis^N_{TV}(Q, I_i) = Dis^N_{TV1}(Q, I_i) + Dis^N_{TV2}(Q, I_i) + \cdots + Dis^N_{TVN}(Q, I_i) \tag{4.3}$$

where $Dis_{NTn}(Q, I_i)$ are the normalized distances of $H_{TVn}, n = \{1, 2, \ldots, N\}$. The global distance used to evaluate the similarity between the query and images in the database is then given by:

$$Dis_G(Q, I_i) = \alpha \times Dis^N_{CV}(Q, I_i) + (1 - \alpha) \times Dis^N_{TV}(Q, I_i) \tag{4.4}$$

where $(0 \leq \alpha \leq 1)$ is a weight parameter that controls the impact of color feature and texture feature in the procedure of image retrieval.

4.5 Descriptor of color texture generated by sparse representation

In this section, sparse representation is introduced into color texture retrieval by proposing a new framework for this application field, in which images are represented

by sparse representation based histogram. Comparing with classical histogram of vectors, in which feature vector is only projected into one partition of vectors, sparse representation based histogram projects feature vector into many partitions of vectors with different weights. In other words, one feature vector will be represented by a few basis vectors instead of one basis vector and this will let the histogram to be a more accurate representation of the images. Details of sparse representation based histogram can be found in Section 2.5.

Two key problems should be considered for applying sparse representation based histogram: feature vectors and dictionary. So multiresolution feature vector and dictionary in wavelet domain are proposed.

4.5.1 Multiresolution feature vectors

Color images are decomposed as described in Section 4.3. The approximation subband W_S^{LLN}, $S \in \{Y, Cb, Cr\}$ is a sub-sampled version of the original image. The detail subbands W_S^{HLn}, W_S^{HLn} and W_S^{HHn}, $S \in \{Y, Cb, Cr\}$ mostly represent the information of local discontinuities of horizontal, vertical and diagonal directions in the image, that's to say the directional information of the image. The feature vectors constructed by the coefficients from each subband are also categorized into two kinds: approximation vector A and detail vector T. A is constructed from W_S^{LLN} whose elements are the coefficients at the same location in each color component. T_n are constructed from nine detail subbands at the same decomposition level in each color component whose elements are the coefficients at the same location in each of the horizontal, vertical and diagonal subbands. Figure 4.3 graphically shows this procedure when $N = 2$, and the definitions of feature vectors are as follow:

$$\begin{aligned} A = [&W_Y^{LLN}(x_a, y_a),\ W_{Cb}^{LLN}(x_a, y_a),\ W_{Cr}^{LLN}(x_a, y_a)] \\ Tn = [&W_Y^{HLn}(x_n, y_n),\ W_Y^{LHn}(x_n, y_n),\ W_Y^{HHn}(x_n, y_n), \\ &W_{Cb}^{HLn}(x_n, y_n),\ W_{Cb}^{LHn}(x_n, y_n),\ W_{Cb}^{HHn}(x_n, y_n), \\ &W_{Cr}^{HLn}(x_n, y_n),\ W_{Cr}^{LHn}(x_n, y_n),\ W_{Cr}^{HHn}(x_n, y_n)] \end{aligned} \tag{4.5}$$

where $n = \{1, 2, \ldots, N\}$ and (x_a, y_a) and (x_n, y_n) indicate the coordinates of the coefficients in approximation subband and detail subband respectively. For N decomposition levels,

we have one set of A and N sets of Tn.

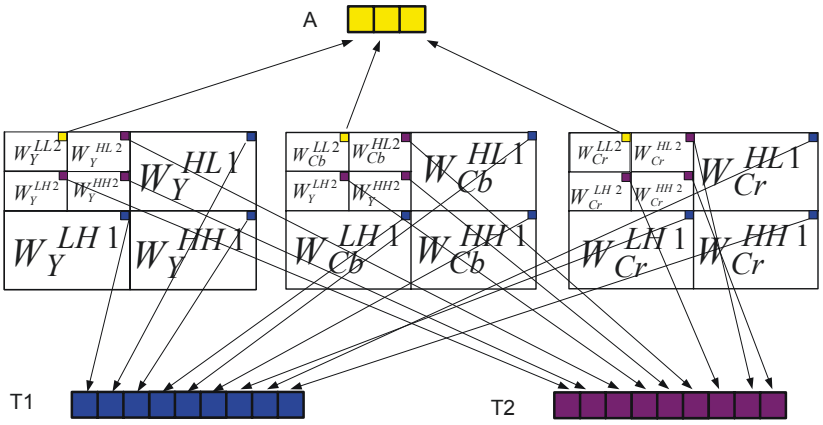


Figure 4.3: Mapping coefficients into multiresolution vectors (N=2)

4.5.2 Dictionaries

As we have $N+1$ kinds of feature vectors, $N+1$ dictionaries should be constructed for sparse representation, which are generated from the corresponding feature vectors of the training images. Random selection of a quarter of images from image database is chosen as training set. There are many methods presented for dictionary learning, but in this work we are not concerned with finding the best possible method. So the simplest choice is using widely used unsupervised learning method: K-means. From giving training set, feature vectors are generated as we described before, and the cluster centers of these training vectors resulted from K-means are used as the dictionary. So we get $\mathbf{D}_A$ and $\mathbf{D}_{Tn}$ respectively, where $n = \{1, 2, \ldots, N\}$, $\mathbf{D}_A \in \mathbb{R}^{3\times K_A}$ and $\mathbf{D}_{Tn} \in \mathbb{R}^{9\times K_{Tn}}$ (K_A and K_{Tn} indicate the number of cluster centers). In the step of performance evaluation, K_A and K_{Tn} are fixed to 400 and 1500 respectively, which were found experimentally.

4.5.3 Similarity measurement

With $N+1$ kinds of feature vectors and $N+1$ dictionaries, $N+1$ sparse representation based histograms $(H_A, H_{T1}, \ldots, H_{TN})$ can be got for one image. We always chose χ^2 distance to measure the similarity between the histogram of query H_Q and the histogram of i^{th} image from the database H_{I_i}. The fused distances of each histogram is used to measure the similarity of the image.

Since each histogram are constructed from different feature vectors, values of distances

range differently. Before using fusion of the distances of different histograms, they should be normalized. Distances can be normalized through the ways described in Section 3.5.5.

The global distance used to evaluate the similarity between the query and images in the database is then given by:

$$Dis_G(Q, I_i) = \beta \times Dis_A^N(Q, I_i) + (1 - \beta) \times Dis_T^N(Q, I_i) \tag{4.6}$$

where$(0 \leq \beta \leq 1)$ is a weight parameter that controls the impact of different histograms in the procedure of image retrieval. And $Dis_A^N(Q, D_i)$ and $Dis_T^N(Q, D_i)$ are the normalized distances of H_A and H_{Tn} respectively, and

$$Dis_T^N(Q, I_i) = Dis_{T1}^N(Q, I_i) + Dis_{T2}^N(Q, I_i) + \cdots + Dis_{TN}^N(Q, I_i) \tag{4.7}$$

where $Dis_{Tn}^N(Q, I_i), n = \{1, 2, \ldots, N\}$ are the normalized distances of H_{Tn}.

4.6 Experimental results

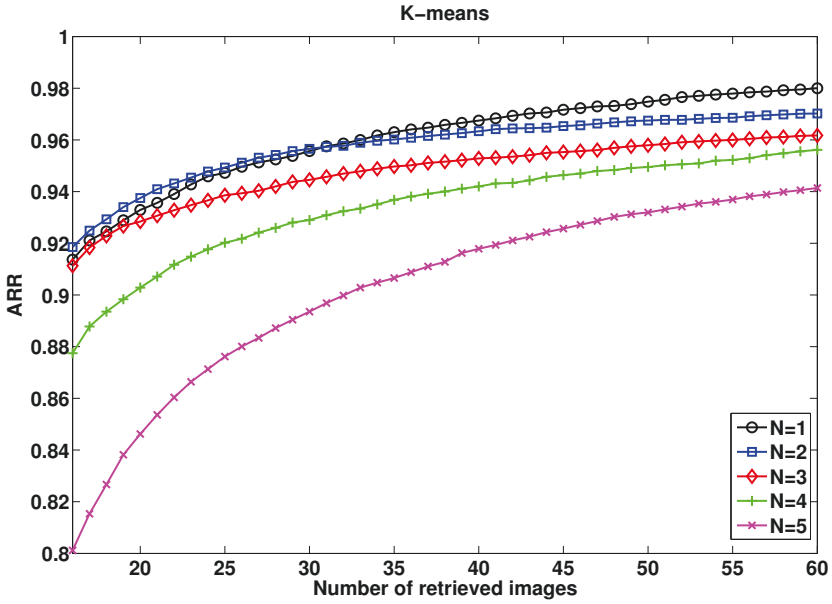
The contents of experiments are listed as: 1) Showing the effect of decomposition level N. 2) Analyzing the experiments and showing some failed retrieval results. 3) Comparing with state-of-the-art methods.

We will evaluate our method on VisTex texture database [85] firstly, similar as the experiments in the Section 3.5.6. For comparison purpose, we evaluate our proposal on two data sets: one is on the classical selection of 40 classes of textures that are used by many literatures about texture retrieval and we have named it 'Small VisTex'. And the other one is on the whole collection of VisTex, that means the selection of 167 classes of texture.

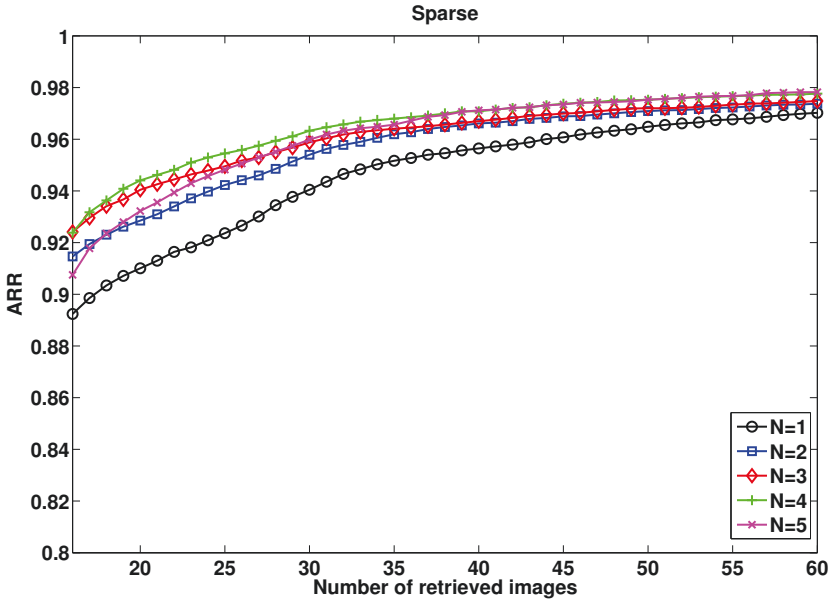
In the retrieval experiments, for all data sets, each subimage in the database is used once as a query. ARR and Precision-Recall are used to evaluate the performance. The relevant images for each query consists of all the subimages from the same original texture.

We should also emphasize that for different α in Equation 4.4 (controlling the impact of color-vector and texture-vector) and β in Equation 4.6 (controlling the impact of proximation vector and detail vector), various ARR can be got because of different impacts of color feature and texture feature or different impacts of approximation feature and detail feature in the process of retrieval. To avoid over-optimization of this parameter for differ-

ent data sets, all the results presented below are the ARR for $\alpha = 0.35$ in Equation 4.4 and $\beta = 0.15$ in Equation 4.6: they assure good ARRs that we can experimentally get in all database, but not only optimization for best ARR in one database.



(a) K-means method



(b) Sparse method

Figure 4.4: ARR according to the number of top matches considered on Small VisTex

4.6.1 Effect of decomposition level

The number of decomposition levels used in DWT affects the retrieval performance of the proposed methods. In order to observe this effect, retrieval are performed for different decomposition levels $N = \{1, 2, 3, 4, 5\}$ and ARR is computed according to different number of similar images retrieved. The ROCs on Small VisTex and whole VisTex are shown in Figure 4.4 and Figure 4.5 respectively.

Table 4.1 shows the numerical comparison between both methods on these two databases. We can conclude that the global performance of sparse method is better than K-means method as the mean value of ARR on different decomposition level of sparse method is higher than that of K-means method while operating with less decomposition levels, K-means method outperforms. This conclusion is affirmed again from the point of view of Precision-Recall pair, as shown in Figure 4.6 and Figure 4.7.

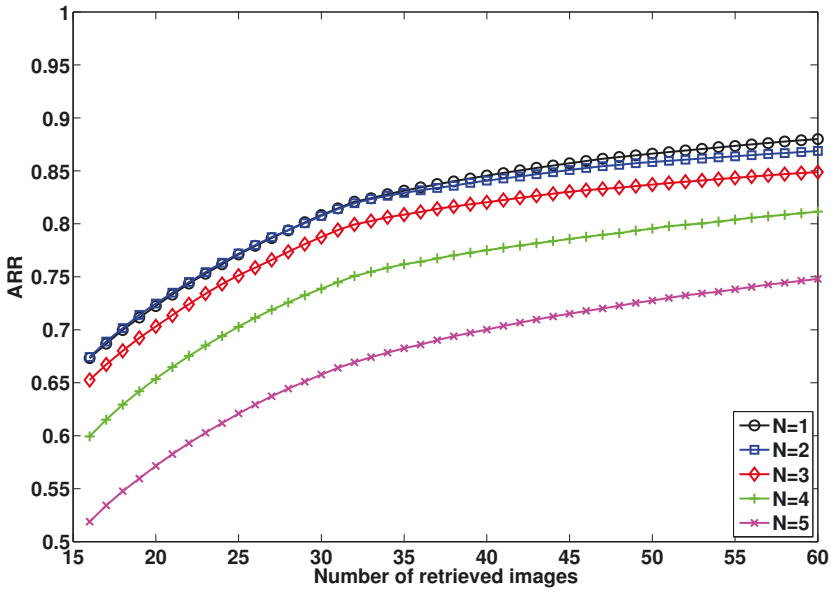
Table 4.1: Comparison of ARR of different decomposition levels [%]

Numbers of Top matches	Decomposition Level	Small VisTex		Whole VisTex	
		K-means	Sparse	K-means	Sparse
Top 16	N=1	91.37	89.24	67.33	67.99
	N=2	91.86	91.46	67.43	69.98
	N=3	91.13	92.41	65.26	69.5
	N=4	87.87	92.38	59.94	69.08
	N=5	80.12	90.75	51.88	66.53
	Average ARR	88.44	**91.25**	62.37	**68.61**
Top 60	N=1	98	97.02	88	86.76
	N=2	97.02	97.37	86.88	87.62
	N=3	96.17	97.48	84.88	87.87
	N=4	95.62	97.76	81.15	88.53
	N=5	94.14	97.82	74.8	87.78
	Average ARR	96.19	**97.49**	83.14	**87.71**

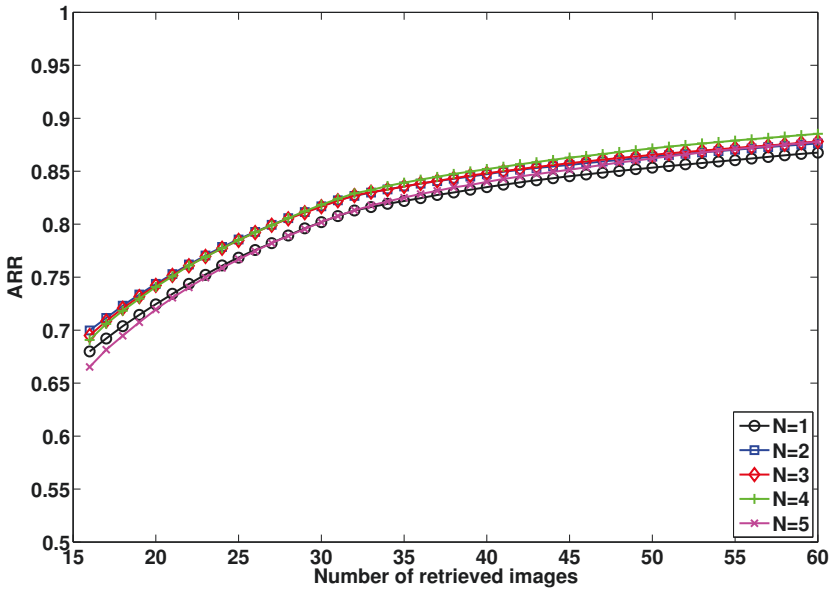
Table 4.2 provides detailed retrieval rate of every texture class on Small VisTex in case of top 16 similar images are retrieved. From this table, we can see that from the point of view of ARR, K-means method is almost invariant to the decomposition levels when $N \leq 3$, and then the performance decreases rapidly when $N \geq 3$. The reason is mainly because when N increases, the amount of coefficients that are used for constructing feature vectors decreases as we said in Section 4.4.1; while sparse method reaches the highest value when $N = 3$, but the variance between ARR of different decomposition levels is small. Thus decomposition level $N = 3$ gives satisfying results for both methods when the size of images is 128×128.

Table 4.2: ARR of each class on Small VisTex (Top 16 Matches)[%]

	K-means					Sparse				
Texture	N=1	N=2	N=3	N=4	N=5	N=1	N=2	N=3	N=4	N=5
bark.0000	**54,30**	**55.86**	**58.98**	**56.64**	**53.52**	86.72	86.72	71.09	87.11	87.11
bark.0006	74.22	76.95	80.86	80.86	69. 14	75.78	75. 78	95.70	89.45	90.23
bark.0008	61.33	64.06	62.50	59.38	46.48	71.09	76.56	73.83	82.81	80.08
bark.0009	**65.23**	**61.72**	**55.08**	**41.41**	**36.72**	60.94	65.63	81.25	75.00	67.58
brick.0001	100.00	100.00	100.00	99.61	93.75	100.00	100.00	100.00	100.00	100.00
brick.0004	100.00	100.00	100.00	99.22	89.06	96.09	98.05	98.44	96.48	91.02
brick.0005	97.27	97.66	94.53	80.47	61.33	98.05	98.83	98.44	94.92	88.67
buildings.0009	100.00	100.00	97.66	95.31	90.63	100.00	100.00	100.00	100.00	100.00
fabric.0000	100.00	100.00	100.00	99.22	97.66	83.20	83.20	84.77	88.28	88.67
fabric.0004	76.95	74.61	74.61	78.52	76.17	**55.47**	**64.45**	**60.94**	**71.09**	**74.22**
fabric.0007	98.05	98.44	96.88	96.48	92.58	100.00	100.00	100.00	100.00	99.22
fabric.0009	99.61	100.00	100.00	97.66	86.72	98.83	99.22	98.44	93.75	87.11
fabric.0011	99.61	100.00	100.00	99.22	89.84	100.00	100.00	99.22	100.00	97.66
fabric.0014	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
fabric.0015	100.00	100.00	99.61	94.14	83.59	99.61	99.61	98.83	100.00	100.00
fabric.0017	98.44	97.27	94.53	93.75	86.33	99.61	97.66	97.27	92.58	92.58
fabric.0018	87.89	90.63	98.44	96.88	96.09	80.08	99.22	99.22	96.48	92.58
flowers.0005	100.00	100.00	100.00	99.61	91.02	100.00	100.00	100.00	99.22	98.44
food.0000	100.00	100.00	100.00	100.00	97.66	100.00	100.00	100.00	100.00	99.22
food.0005	93.36	91.02	88.67	83.20	78.13	99.61	100.00	100.00	100.00	100.00
food.0008	100.00	100.00	99.22	89.06	67.19	100.00	100.00	100.00	100.00	100.00
grass.0001	93.75	96.48	91.02	91.80	81.64	72.66	79.69	71.48	81.25	77.73
leaves.0008	94.92	93.36	84.77	75.78	63.28	97.66	99.61	98.83	96.09	94.14
leaves.0010	99.22	97.66	98.44	92.19	76.56	99.22	100.00	100.00	100.00	100.00
leaves.0011	100.00	100.00	98.83	93.36	81.25	100.00	100.00	100.00	99.61	95.70
leaves.0012	58.98	74.22	85.94	83.98	66.80	**50.00**	**50.00**	**49.61**	**50.00**	**50.00**
leaves.0016	94.53	96.48	91.80	78.52	65.23	88.67	95.31	87.11	91.80	82.42
metal.0000	100.00	99.22	100.00	99.61	94.53	96.88	98.83	91.80	95.31	91.02
metal.0002	100.00	100.00	100.00	100.00	96.88	100.00	100.00	100.00	99.61	97.27
misc.0002	100.00	100.00	100.00	100.00	99.61	100.00	100.00	100.00	100.00	100.00
sand.0000	100.00	100.00	100.00	99.61	96.88	100.00	100.00	100.00	99.61	98.05
stone.0001	96.88	92.97	83.98	72.66	60.55	95.31	95.70	96.88	94.53	92.97
stone.0004	71.48	82.42	91.41	93.36	92.97	60.94	84.77	90.23	88.67	78.13
terrain.0010	90.23	80.86	70.70	65.63	59.77	75.39	78.91	96.88	94.14	96.09
tile.0001	83.20	74.61	69.14	63.67	59.77	77.73	74.61	71.09	73.44	77.34
tile.0004	100.00	100.00	100.00	93.36	79.30	100.00	100.00	100.00	99.61	98.83
tile.0007	77.73	93.75	98.83	94.53	89.45	73.05	79.30	98.83	88.67	87.89
water.0005	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
wood.0001	91.02	84.77	78.91	71.09	56.64	77.34	76.95	86.33	75.78	78.13
wood.0002	96.48	99.22	100.00	100.00	100.00	99.61	100.00	100.00	100.00	100.00
ARR	91.37	91.86	91.13	87.74	80.12	89.24	91.46	92.41	92.38	90.75



(a) K-means method

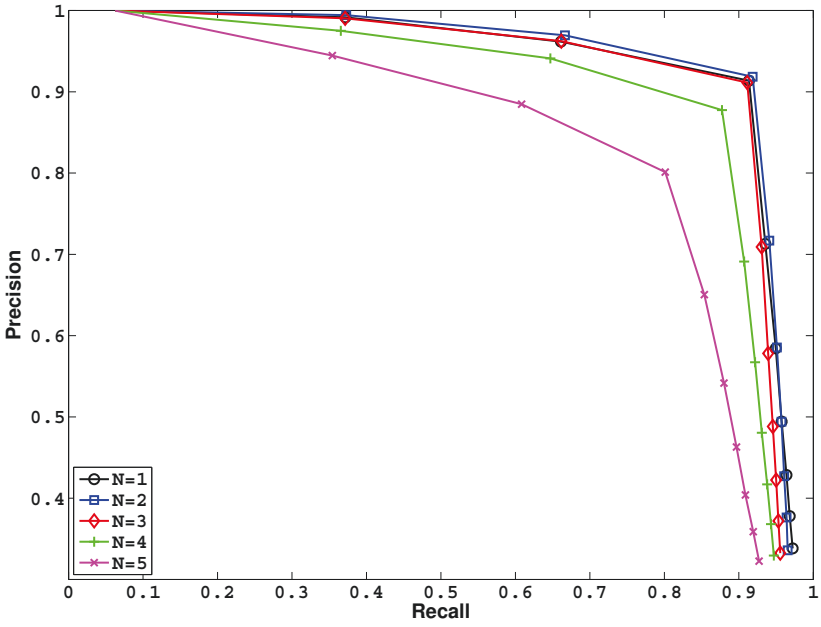


(b) Sparse method

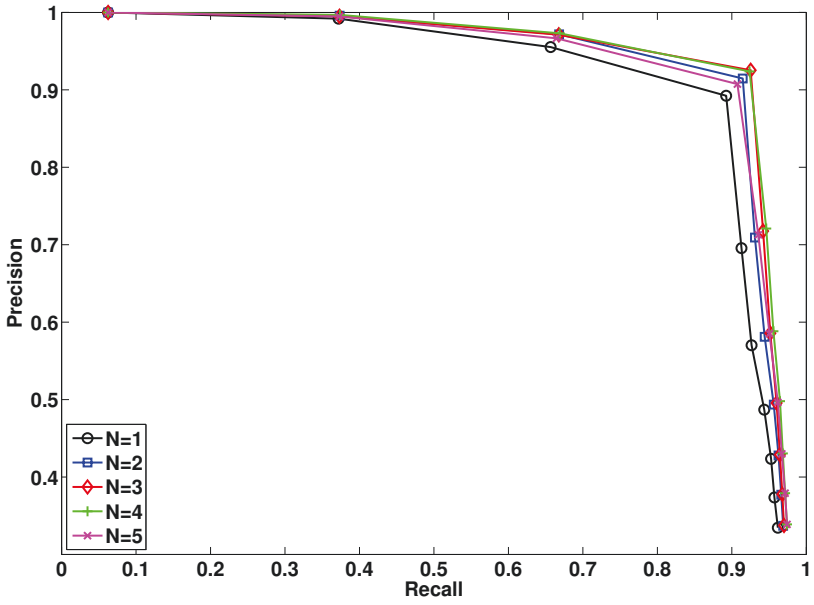
Figure 4.5: ARR according to the number of top matches considered on Whole VisTex

4.6.2 Examples of failed retrieval

Although ARR of both methods are rather high (around 90%). there are also some classes of textures get very low retrieval rate (around 50%). In K-means method, they are 'Bark.0000' and 'Bark.0009', and in sparse method, 'Fabric.0009' and 'Leaves.0012',



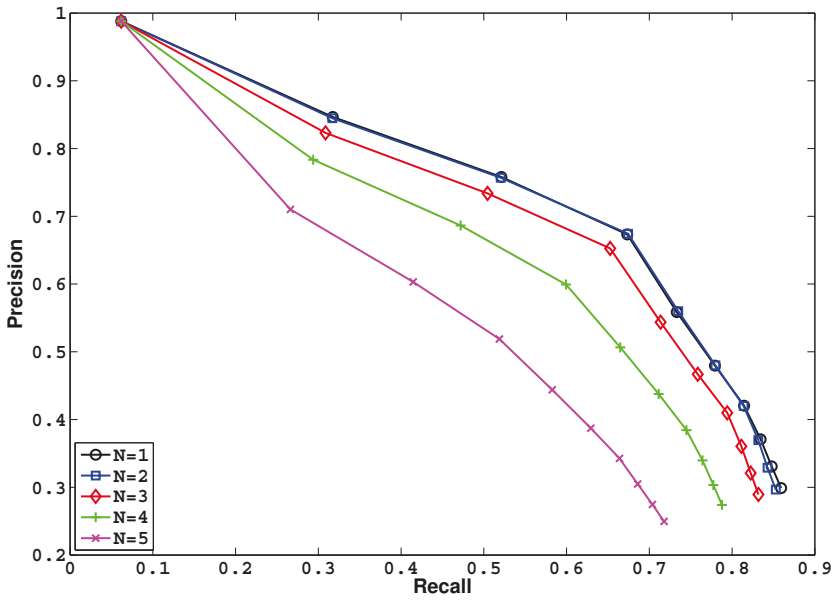
(a) K-means method



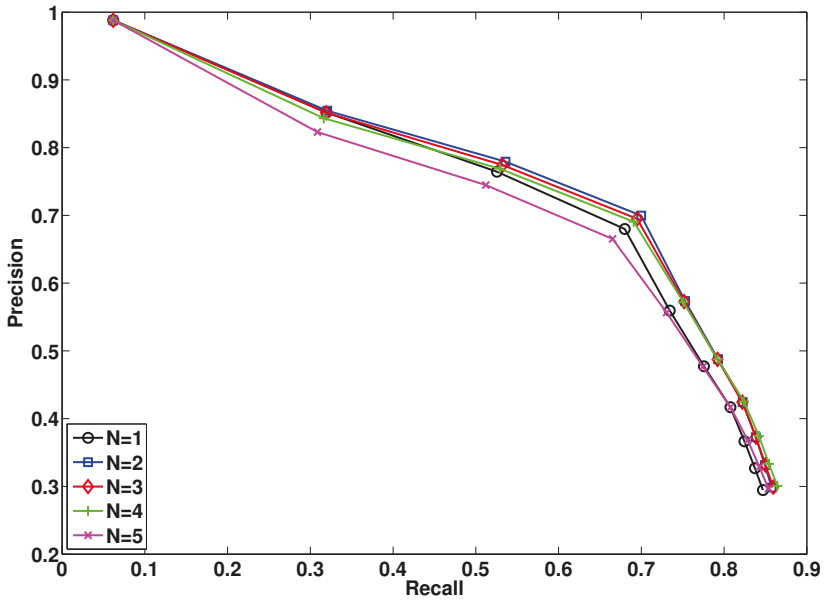
(b) Sparse method

Figure 4.6: The Precision-recall pair on Small VisTex

as marked bold in the Table 4.2. Here we present and analyse some failed examples of retrieval when one subimage of these texture are used as query. Figure 4.8 shows the results of retrieved images when the query image is one of the subimage of 'Bark.0000'



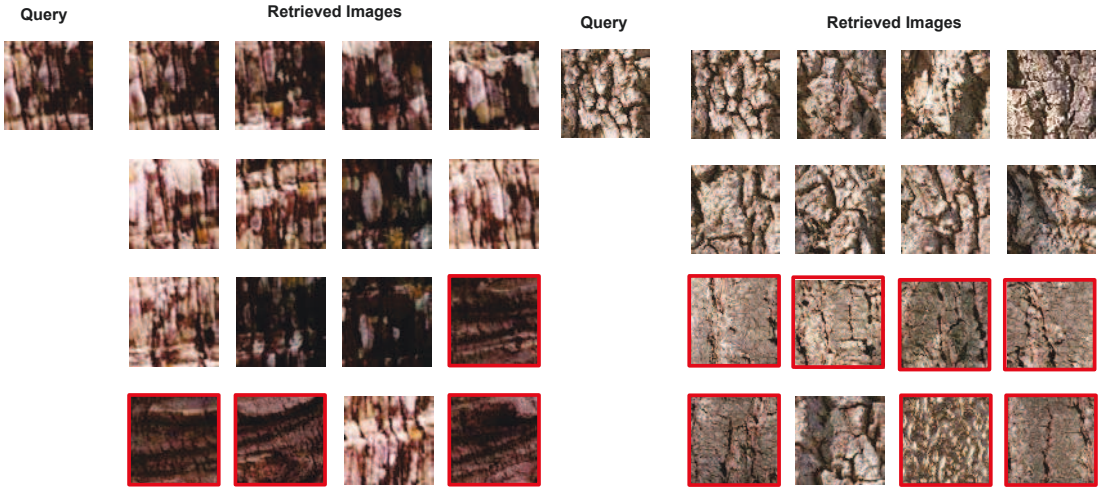
(a) K-means method



(b) Sparse method

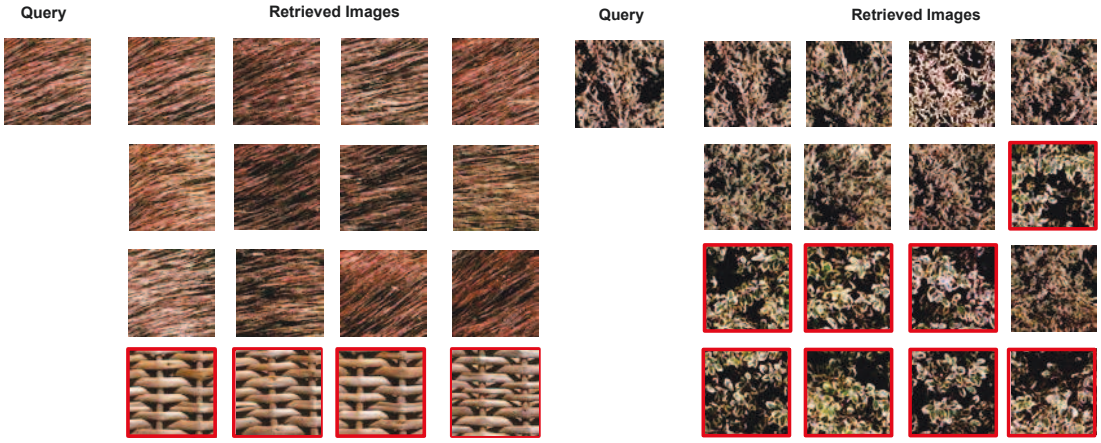
Figure 4.7: The Precision-recall pair on Whole VisTex

and of 'Bark.0009' for K-means method at the decomposition level $N = 3$. And Figure 4.9 shows the results of retrieved images when the query image is one of the subimage of 'Fabric.0004' and of 'Leaves.0012' for sparse method at the decomposition level $N = 3$. The wrong retrieved images are marked in red.



(a) Results for one subimage of 'Bark.0000' (b) Results for one subimage of 'Bark.0009'

Figure 4.8: Retrieved images of top 16 matches by K-means method



(a) Results for one subimage of 'Fabric.0004' (b) Results for one subimage of 'Leaves.0012'

Figure 4.9: Retrieved images of top 16 matches by sparse method

From these four images, we can indicate that, although the failed retrieved images are not good results according to the principle of performance evaluation, they are really visually similar to the query image or have the similar directional information with the query image.

4.6.3 Comparison with state-of-the-art

Finally for a more general evaluation, we give the comparison between our proposals and referred methods including state-of-the-art methods. Besides Small VisTex and whole

VisTex, two new databases are also chosen for comparison with state-of-the-art methods: ALOT [101] and STex [102].

ALOT is a color image collection of 250 textures, in which every texture is recorded under varied viewing angle, illumination angle, and illumination color. Some examples of textures of this database is shown in Figure 4.10. For comparison purpose, only the textures captured under c1l1 condition are selected, which means that images are captured by camera 1 under illumination condition 1. More details can be found in [101]. In the experiments, the 384×256 color version of texture are divided into 16 non-overlapping subimages (96×64), thus creating a database of 4000 images belonging to 250 classes.

Figure 4.10: Examples of textures from ALOT database

STex is a large collection of 476 color texture image that have been captured around Salzburg, Austria. Some examples of textures of this database is shown in Figure 4.11. In the experiments, the 512×512 color version of texture are divided into 16 non-overlapping subimages (128×128), thus creating a database of 7616 images belonging to 476 classes.

As we said before, decomposition level N has the effect on the retrieval performance, and some methods reported their results with given decomposition levels, but others do

Figure 4.11: Examples of textures from STex database

not. So for objective comparison, N is set to 3 when compare with methods presented in [87, 94, 95], and set to 2 when [88] is compared. For those that we didn't know the decomposition levels or non-wavelet methods, the best ARRs are compared.

Table 4.3 and table 4.4 present the comparative experimental results on small VisTex. We repeat the principles of referred methods. 'MGG' is a method presented in [94], in which wavelet coefficients are modeled by Multivariate Generalized Gaussian (MGG) jointly in each color components of texture and geodesic distance is used for similarity measure. 'GCG' represents the method introduced in [87], in which images are decomposed by complex wavelet transform, coefficients in each subband are modeled by Gaussian Copula with Gamma (GCG) distribution and statistics and marginal parameters of each band form feature vectors. And L_1 distance is used to measure the similarity. 'Student-t' is seen as one state-of-the-art method presented in [88], in which student-t distribution are used to model the color cue and spatial dependencies of wavelet coefficients and geodesic distance is used again for similarity measure. 'GCWD' is another state-of-the-art approach presented in [95], in which wavelet coefficients are modeled by several distributions, but

Gaussian Copula with Weibull distribution (GCWD) outperforms. It can be observed that sparse method outperforms referred methods including state-of-the-art methods while K-means method has a better performance than almost all referred methods except 'MGG' method.

Table 4.3: ARR on Small VisTex (N=3)

Method	MGG [94]	GCG [87]	GCWD [95]	K-means	Sparse
ARR(%)	91.2	85.83	89.5	91.13	**92.41**

Table 4.4: ARR on Small VisTex (N=2)

Method	Student-t [88]	K-means	Sparse
ARR(%)	89.65	**91.86**	91.46

Table 4.5 presents the retrieval performance on the whole VisTex, ALOT and STex database. We note that Sparse method still outperforms in these three large databases and with obviously improvements in ALOT and STex, while K-means method outperforms in ALOT but only give same level of performance with referred methods in STex and ranks in the third position in whole VisTex.

Table 4.5: ARR on whole VisTex, ALOT and STex (N=3)[%]

	MGG [94]	GCWD [95]	K-means	Sparse
VisTex	69.3	63.8	67.43	**69.98**
ALOT	49.3	54.1	**59.69**	58.08
STex	71.3	70.6	69.85	**78.07**

4.6.4 Conclusion of experiments

From above experiments, we can get two conclusions as follows: 1) Decomposition level N should be chosen appropriately according to the method. 2) Compared with state-of-the-art methods, K-means method can not always get the better performance while sparse method always outperforms. In consideration of less amount of wavelet coefficients used in K-means method, it is not a bad choice for color texture retrieval. However, as it is pointed out in [95], computing the similarity between histograms is less expensive in terms of arithmetic operations than computing any of similarity measure that used in the framework of probabilistic image retrieval.

4.7 Conclusion

In this chapter, we have detailed two proposals for color texture retrieval in wavelet domain.

The first proposal is in the context of extracting color and texture features separately: color features are extracted by the coefficients of approximation subband of DWT in luminance and chrominance components of images and texture features are constructed by the coefficients of detail subbands of DWT in luminance components . This proposal brings one contribution: with the proposed multiresolution texture-vectors and color-vector, only at most 50% wavelet coefficients need to be processed for constructing feature vectors.

The second proposal is in the context of extracting color and texture features jointly: multiresolution features are extracted jointly both from luminance and chrominance components of color texture. The main contribution of this proposal is introducing sparse representation into color texture retrieval by proposing sparse representation based histogram. This contribution leads to two advantages: 1) The sparse representation based histogram is more accurate as feature descriptor which leads to higher accuracy in the color texture retrieval. 2) This approach for color texture retrieval has less expense in computing load than the framework of probabilistic color texture image retrieval in the aspect of computing the similarity between feature descriptors.

Experimental results, got on four data sets: classical selection of 40 textures of VisTex, the whole VisTex, ALOT and STex, confirmed the contributions of our two proposals by comparing with state-of-the-art approaches in wavelet domain.

5

Conclusion and perspective

This thesis has introduced, analyzed and studied CBIR in transform domain. This work mainly focus on image descriptor extracted from DCT and DWT that are widely used for compressing images and videos.

The background and some fundamental concepts related to CBIR and our proposals have been firstly introduced. In general, there are two categories of features used for image retrieval: intensity-based (color and texture) and geometry-based (shape). We have focused on intensity-based one. We have analyzed the properties of coefficients of DCT and DWT and have shown the possibilities to construct feature vectors directly from transformation coefficients. Then the construction of histogram was introduced. K-means and sparse representation used for constructing histograms were also presented. We have proposed a new kind of histogram: sparse representation based histogram, in which one target vector is represented by a few basis vectors instead of by one basis vector in the classical histogram. This leads to a more accuracy representation of target vectors. And then the similarity measurement, especially the distances between histograms have been introduced. Finally, the evaluation methods for CBIR performances have been presented and chosen.

We have proposed one improved approach and two new approaches in DCT domain. The improved approach is based on a method in which the histogram of AC-Patterns and that of DC-Patterns are used as feature descriptor. AC-Pattern consists of the AC coefficients extracted from one DCT block. Two aspects of improvements were proposed: zigzag scan used for arranging coefficients in AC-Pattern and merging adjacent patterns. Considering the properties of DCT coefficients, two kinds of feature vectors were proposed: Sum-Pattern and Texture-Pattern. Both feature vectors are 3-D vectors that could repre-

sent the directional information of DCT block efficiently and can be applied both on face recognition and texture retrieval. Finally, Color-Pattern was proposed and used in conjunction with Texture-Pattern for color texture retrieval. Experimental results on widely used face database (ORL, GTF and FERET) and popular texture database (VisTex) have demonstrated the efficiency of our approaches.

We have proposed two approaches for color texture retrieval in wavelet domain. The first one operates in the context of constructing color and texture features separately: color features are extracted by the coefficients of approximation subband of DWT and texture features are constructed by the coefficients of detail subbands of DWT in luminance components. The main advantage of this proposal is that it only requires at most 50% wavelet coefficients to be processed. The second approach is in the context of using the color-texture features jointly. This contribution leads to two advantages: one is that one feature vector will be represented by a few basis vectors instead of one basis vector in sparse representation based histogram. This will let the histogram to be more accurate for representing the images; another one is that this approach is much less expensive in computing load in the aspect of computing the similarity between feature descriptors. Experiments have been executed on four widely used color texture database: Small VisTex, whole VisTex, ALOT and STex. The effect of decomposition level and failed retrieval examples were also analyzed. Furthermore, comparison with several state-of-the-art has also been done with success. All these results confirmed the contribution of these two proposals.

Future perspective works will focus on following aspects:

1. Extending the sparse representation based histogram to DCT domain. To do this, two problems should be solved: on one hand, appropriate feature vectors should be constructed; on the other hand, corresponding dictionaries should be found.
2. Verifying the capability of multiresolution feature vectors in DWT domain for retrieving multiresolution images. In the experiments of evaluation, images in the database are supposed to have the same resolution. As multiresolution feature vectors have been defined, it should be interesting to see the performance of retrieval when the images in the database have different resolutions.
3. Finding a more compact dictionary. In current approaches, one kind of feature vectors needs to one corresponding dictionary. But the ideal compact dictionary satisfies two conditions: the size of dictionary should be the smaller the better and

the number of dictionaries also should be smaller. So the best choice is to build only one dictionary with reasonable size that could be suitable for constructing histograms of all kinds of multiresolution feature vectors.

4. Integrating more features in the approach based on sparse representation based histogram. Only global texture and color features are considered in proposed approaches, but other kinds of features could be also used separately or jointly: shape features or local features.

A

Appendix : Résumé étendu en français

Chapitre 1 : Introduction

L'accroissement des bases de données d'images numériques augmente la quantité d'information disponible pour les utilisateurs. La difficulté d'utiliser efficacement ces informations s'accroît également. La recherche d'images s'utilise pour parcourir, rechercher et récupérer facilement les données. L'objectif de la recherche d'images est de fournir un accès facile à l'image dans les bases des données. Deux méthodes existent dans ce domaine : la premiere basée sur le texte et la seconde basée sur le contenu.

La méthode basée sur le texte utilisant l'indexation par les mots-clés du texte est la méthode de recherche d'images la plus courante. La technique basée sur le texte est précise et efficace pour trouver les images annotées. Cependant, trois difficultés principales surviennent. Premièrierement, l'explosion de la quantité d'informations nécessite des annotations manuelles laborieuses. Deuxièmement, la richesse du contenu des images et la subjectivité de l'annotation manuelle peuvent conduire à des annotations différentes sur une même image et induire à une inadéquation irréversible dans les processus de recherche. Enfin, les annotations doivent être faites des langues différentes, accroissant ainsi la difficulté de recherche.

Pour résoudre ces trois problèmes, la méthode de recherche d'images par le contenu (CBIR) est proposée depuis le début des années 1990. Elle est basée sur un nouveau mode de recherche d'images, dans lequel les images sont indexées par leurs propres contenus visuels. Le but principal du CBIR est d'obtenir les images qui sont visuellement similaires à la requête présentée. Le schéma général de CBIR est illustré sur la Figure A.1.

Son principe est l'extraction de caractéristiques, un processus de transfert de l'image

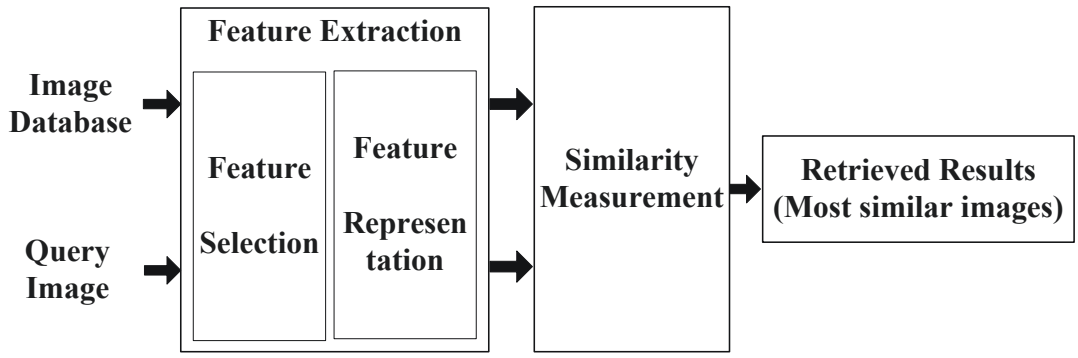


FIGURE A.1 – La recherche d'images par le contenu

d'entrée à ensemble des caractéristiques (également nommé vecteurs caractéristiques). Il existe deux types principaux de vecteurs caractéristiques en recherche d'images : le premier basé sur l'intensité (couleur et texture), le deuxième sur la géométrie (forme). La représentation des vecteurs caractéristiques est aussi appelée descripteur des caractéristiques. Un descripteur peut être global ou local. Un descripteur global utilise des caractéristiques visuelles de l'image entière alors qu'un descripteur local utilise des caractéristiques visuelles limitées à un voisinage local.

Une fois l'extraction de caractéristiques terminée, la distinction des images similaires doit être prise en compte. Les technologies peuvent se diviser en la similarité basée région, la similarité globale, ou la combinaison des deux.

Cette thèse se concentre sur la recherche d'images basée sur l'intensité. Même si le concept pour distinguer l'écart sémantique des caractéristiques entre bas niveau et haut niveau est encore un problème dans un système CBIR, la similarité visuelle peut être plus critique que la similarité sémantique pour certaines applications. L'extraction de caractéristiques de couleur et de texture est toujours un problème et les approches existantes ne suffisent pas pour le résoudre, surtout lorsqu'elle est appliquée sur différents types de base de données. Notre idée est d'essayer de trouver une approche qui peut extraire directement les caractéristiques de couleur et de texture à partir des domaines transformés et d'utiliser leur combinaisons pour effectuer la recherche d'image. Ce travail est motivé par le fait que la majorité des images sont stockées dans un format compressé et que les technologies de compression utilisant différents types de transformations.

Ce résumé français introduit les contenus principaux de la thèse. Il s'articule comme suit : les concepts fondamentaux du CBIR ainsi que les théories utilisées dans les approches proposées sont introduits dans le chapitre 2. L'approche dans le domaine de la transfor-

mée en cosinus discrète (Discrete Cosine Transform, DCT) pour images compressées avec JPEG est présentée au chapitre 3, dans ce chapitre on améliore les méthodes existantes et on propose deux nouvelles approches. Enfin le chapitre 4 détaille les approches dans le domaine de la transformée en ondelettes discrète (Discrete Wavelet Transform, DWT) pour JPEG2000 où deux méthodes sont également proposées. La conclusion et les perspectives sont présentées dans le chapitre 5.

Chapitre 2 : Concepts fondamentaux

Ce chapitre introduit les concepts fondamentaux concernant le CBIR dans le contexte des domaines transformés. Les transformées DCT et DWT sont d'abord introduites. À partir des coefficients de ces deux transformations, des vecteurs caractéristiques sont construits. Puis deux approches sont utilisés pour la création d'histogrammes : le regroupement des données et la représentation parcimonieuse sont aussi introduits. Enfin, la mesure de similarité entre les histogrammes et l'évaluation de la performance pour le CBIR sont détaillées.

2.1 La transformée en cosinus discrète

La DCT 2-D pour un bloc de taille $N \times M$ est donnée par :

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \tag{A.1}$$

pour $u = 0, 1, 2, \ldots, N-1, v = 0, 1, 2, \ldots, M-1$, et la transformation inverse est définie comme :

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \alpha(u)\alpha(v)C(u,v) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \tag{A.2}$$

pour $x = 0, 1, 2, \ldots, N-1, y = 0, 1, 2, \ldots, M-1$. Et $\alpha(o), o \in \{u, v\}$ est définie comme :

$$\alpha(o) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad o = 0 \\ \sqrt{\frac{2}{N}} & for \quad o \neq 0 \end{cases} \tag{A.3}$$

Dans Equation A.2,

$$B(u,v) = \alpha(u)\alpha(v) \cos[\frac{\pi(2x+1)u}{2N}] \cos[\frac{\pi(2y+1)v}{2M}] \tag{A.4}$$

sont les images de base de la DCT 2-D. Avec ces images de base, la transformation DCT inverse peut être réécrite comme :

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} C(u,v)B(u,v) \tag{A.5}$$

Dans cette équation, l'image $f(x, y)$ peut être traitée comme la somme pondéré des images de base $B(u, v)$, avec $C(u, v)$ représentant le poids. Les images de base pour $N = M = 8$ sont présentées sur la Figure A.2. Pour l'illustration, chacune est représentée comme une image en niveaux de gris : plus la valeur de ses points est petite, plus le pixel est sombre ; plus la valeur est grande, plus le pixel est clair. Nous pouvons noter que les différentes images de base présentent une augmentation progressive de la fréquence dans le sens vertical et horizontal. Le coefficient DC en haut à gauche est la valeur moyenne du bloc de pixels et les coefficients AC peuvent être considérés comme la description progressive de fréquences dans le sens vertical et horizontal. Considérant l'explication ci-dessus, les coefficients $C(0, v), v = (1, 2, \ldots, 7)$ représentent les informations des structures horizontales de l'image, et les coefficients $C(u, 0), u = (1, 2, \ldots, 7)$ représentent les informations des structures verticales, les coefficients $C(u, v), u = v = (1, 2, \ldots, 7)$ représentent les informations diagonales.

Nos travaux s'appuient sur les coefficients des blocs DCT 4×4 parce qu'ils donnent des informations plus perceptuelles que celles des blocs 8×8 DCT. Une méthode efficace d'extraction de blocs 4×4 à partir des blocs DCT 8×8 est proposée.

2.2 La transformée en ondelettes discrète

Une ondelette est définie comme :

$$\mathbf{f} \xrightarrow{\mathbf{H}_1} (\mathbf{a^1} \mid \mathbf{d^1}) \tag{A.6}$$

qui donne d'un signal discret $\mathbf{f}$ la moyenne $\mathbf{a^1}$ et les détails $\mathbf{d^1}$, où $\mathbf{a}^1 = (a_1, a_2, \cdots, a_{N/2})$, $\mathbf{d}^1 = (d_1, d_2, \cdots, d_{N/2})$, avec

$$d_m = \mathbf{f} \cdot \mathbf{W}_m^1 \tag{A.7}$$

$$a_m = \mathbf{f} \cdot \mathbf{V}_m^1 \tag{A.8}$$

où $m = 1, 2, \ldots, N/2$, $\mathbf{V}_m^1$ et $\mathbf{W}_m^1$ sont respectivement le signal d'échelle et le signal d'ondelettes. Avec différentes définitions des signaux d'échelle et d'ondelettes, différentes transformées en ondelettes peuvent être obtenues. Parmi eux, l'ondelette Cohen-Daubechies-Feauveau 9/7 (CDF 9/7) est largement utilisée.

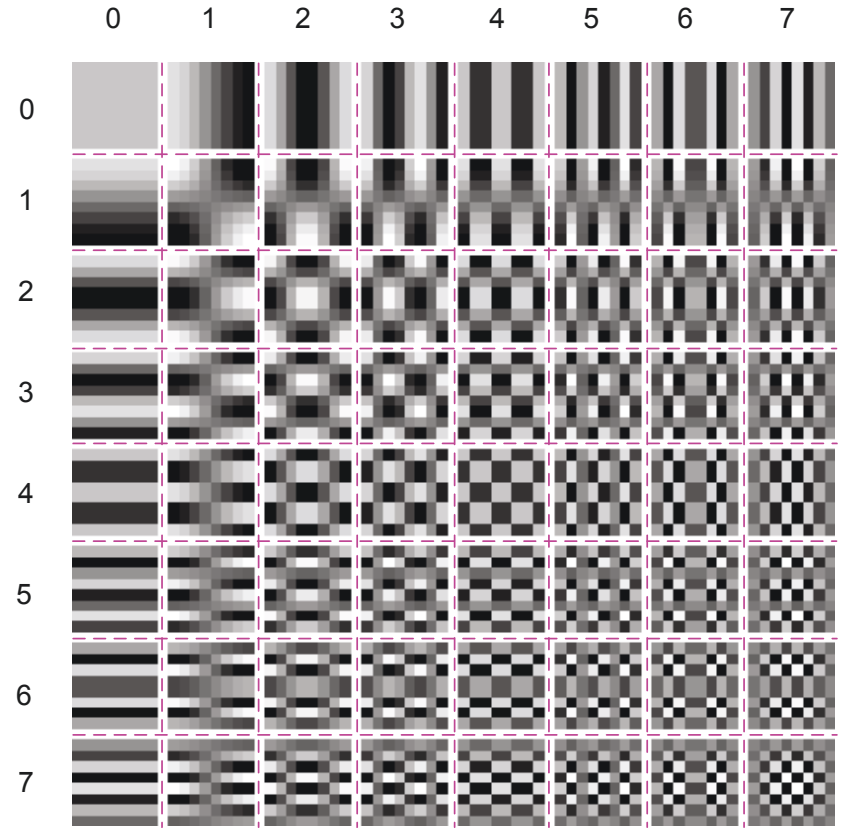


FIGURE A.2 – Fonction de base de DCT 2-D (N=M=8)

Une transformée en ondelettes 2-D d'une image $\mathbf{f}$ au niveau 1 peut être obtenue par une transformation en ondelettes 1-D de niveau 1, sur chaque ligne de $\mathbf{f}$ pour produire une nouvelle image, puis une transformée en ondelettes 1-D effectuée sur chacune des colonnes de cette image. Le niveau 1 d'une décomposition en ondelettes d'une image $\mathbf{f}$ peut être symbolisé comme suit :

$$\mathbf{f} \mapsto \begin{pmatrix} \mathbf{a}^1 & \mathbf{h}^1 \\ \mathbf{v}^1 & \mathbf{d}^1 \end{pmatrix} \tag{A.9}$$

où les sous-images $\mathbf{h}^1$, $\mathbf{d}^1$, $\mathbf{a}^1$ et $\mathbf{v}^1$ ont M/2 lignes et N/2 colonnes.

La sous-image $\mathbf{a}^1$ est créée par les moyennes calculées le long des lignes de $\mathbf{f}$ puis le calcul des le moyennes le long des colonnes. Il s'agit donc d'une version basse résolution de l'image $\mathbf{f}$. La sous-image $\mathbf{h}^1$ est créée par les évolutions calculées le long des lignes de l'image $\mathbf{f}$ puis par le calcul des variations le long des colonnes. Par conséquent, les variations le long d'une colonne sont capables de détecter les contours horizontaux dans l'image. La sous-image $\mathbf{v}^1$ est similaire à la sous-image $\mathbf{h}^1$, sauf que les rôles des variations horizontales

et verticales sont inversés. La sous-image $\mathbf{d}^1$ exprime les fluctuations des caractéristiques en diagonale, car elle est construite à partir des variations le long des lignes et des colonnes.

2.3 Histogramme

L'histogramme est choisi comme le descripteur caractéristique des images dans nos méthodes. En représentation de caractéristiques, un histogramme $\{h_i\}$ est la transformation d'un ensemble de vecteurs de dimension d à un ensemble de valeurs non négatives réelles. Suite à ce transformation, ces vecteurs sont représentés par des classes, indexées i, ce qui correspond à des partitions fixes des vecteurs. Les valeurs réelles associées sont une mesure de la masse des vecteurs qui tombent dans les partitions correspondantes.

2.4 Partitionnement de données

Le partitionnement de données regroupe des objets de sorte que les similarités entre objets d'un même groupe soient élevées, tandis que les similarités entre objets de groupes différents restent faibles. Le plus populaire et le plus simple des algorithmes de partitionnement de données est l'algorithme K-means. L'algorithme K-means nécessite trois paramètres a priori : le nombre de partitions K, les centres initiaux des partitions, et la distance métrique.

2.5 Représentation parcimonieuse

La représentation parcimonieuse est une représentation modélisant les données par une combinaison linéaire d'un petit nombre d'éléments de données. Les éléments sont souvent choisis à partir d'un dictionnaire sur-complète qui est une collection d'éléments dont le nombre est supérieur à la dimension des éléments.

La représentation parcimonieuse peut être écrite comme le problème du "*Lasso*" avec des contraintes positives :

$$\begin{aligned} \arg\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X}-\mathbf{DC}\|_{\ell_2} + \lambda\|\mathbf{C}\|_{\ell_1} \\ s.t.\ \mathbf{D}\succeq 0, \mathbf{C}\succeq 0. \end{aligned} \tag{A.10}$$

où $\mathbf{X}$ est la matrice de données dont les colonnes sont des vecteurs caractéristiques, $\mathbf{D}$ est la matrice de base et les colonnes de $\mathbf{D}$ sont des vecteurs de base, $\mathbf{C}$ est la matrice de

coefficients par laquelle un vecteur peut être représenté par une combinaison linéaire avec les vecteurs de base, λ contrôle le compromis entre l'exactitude et la parcimonie.

Dans ce contexte, un vecteur $\mathbf{x}$ dans la matrice de données $\mathbf{X}$ peut être représenté par un nombre réduit de vecteurs de base. Ensuite, l'histogramme basé sur la représentation parcimonieuse des données de la matrice $\mathbf{X}$ est défini comme :

$$h_j = \sum_{i=1}^{N} C_{ij} \tag{A.11}$$

où C_ij est un coefficient de $\mathbf{C}$ et h_j indique la valeur du $j^{\text{ème}}$ coefficient de l'histogramme. De cette façon, les valeurs des coefficients h_j représentent le poids total de vecteurs de base dans la représentation parcimonieuse de la matrice de données.

2.6 Mesure de similarité

La similarité entre deux images est mesurée par la distance entre les descripteurs de caractéristiques des images. Les images similaires ont alors une distance plus faible entre descripteurs ou un score de similarité plus élevé.

Les mesures de similarité entre les histogrammes se classent en deux catégories : la mesure de similarité coefficient à coefficient et la mesure de similarité croisée entre coefficients. Pour les deux, la distance L_1 et la distance χ^2 sont choisies dans nos approches. Ces deux distances sont définies par :

$$d_{L_1}(H_Q, H_D) = \sum_{i=1}^{N} |H_Q(i) - H_D(i)| \tag{A.12}$$

$$d_{\chi^2}(H_Q, H_D) = \sum_{i=1}^{N} \frac{\left(H_Q(i) - H_D(i)\right)^2}{H_Q(i) + H_D(j)} \tag{A.13}$$

où H_Q et H_D sont les histogrammes respectifs de l'image requête et de l'image dans la base de données.

2.7 Evaluation des performances

Pour évaluer et comparer les algorithmes de CBIR, une évaluation de leurs performances est nécessaire.

a) La mesure d'évaluation le plus couramment utilisé pour le CBIR inclut la *précision* et le *rappel*. La *précision* indique l'exactitude de la recherche se définit comme le rapport

du nombre d'images pertinentes récupérées sur le nombre total des images extraites. *Le rappel* indique la capacité de récupération des images pertinentes à partir de la base de données. Il se définit comme le rapport du nombre d'images pertinentes récupérées sur le nombre total d'images pertinentes dans la base de données :

$$\begin{aligned} Precision &= \frac{q}{q+s} \\ Rappel &= \frac{q}{q+t} \end{aligned} \tag{A.14}$$

où q est le nombre d'images pertinentes récupérées, s le nombre d'images récupérées non-pertinentes, t le nombre d'images pertinentes non-récupérées de la base de données.

b) Le *taux de moyen récupération* (Average Retrieval Rate, ARR) est souvent utilisé dans la littérature pour la recherche de texture. Le *taux de récupération* (Retrieval Rate, RR) pour une requête se définit comme le percentage du nombre d'images pertinentes récupérées sur le nombre total d'images pertinentes dans la base de données, observées dans les K premières images extraites :

$$RR = rappel = \frac{q}{q+t} \tag{A.15}$$

ARR se définit comme la valeur moyenne de l'ensemble des taux de récupération des K premières images trouvées a chaque requête. Évidemment, ARR est lié au nombre d'images récupérées. Ainsi, il est alors possible de construire la courbe de fonctionnement caractéristique de récepteur (Receiver Operating Characteristic, ROC) en traçant ARR suivant K.

c) Le *taux d'égale erreur* (Equal Error Rate, EER) est souvent utilisé pour évaluer la performance de la reconnaissance des visages. Lorsque la reconnaissance est effectuée, la similarité entre les images doit être observée. Les images sont considérées comme similaires si la distance entre leurs descripteurs caractéristiques est inférieure à un seuil donné. Si l'on considère une image requête appartenant à la classe A, deux événements peuvent se produire : d'une part, elle peut être reconnue à juste titre, d'autre part, elle peut être rejetée à tort de la classe A. Dans cette dernière situation un ratio est défini et appelé taux de faux rejet (False Rejected Rate, FRR). En revanche, lors de l'examen d'une image

requête n'appartenant pas à la classe A, si on la compare avec les images de la classe A, elle peut être rejetée à juste titre, ou elle peut être accepté à tort dans la classe A. Alors dans cette dernière situation un ratio est défini comme le taux de fausses acceptations (False Accept Rate, FAR). Ces deux taux varient en fonction du seuil. Lorsque FRR et FAR prennent des valeurs égales, un EER est obtenu. Quand cette valeur EER est faible, la performance du système est jugée bonne, car le taux d'erreur totale correspond à la somme des FAR et FRR.

2.8 Conclusion

Dans ce chapitre, quelques concepts fondamentaux utilisés dans nos approches en CBIR sont introduits.

Tout d'abord, nous avons introduit deux transformations couramment utilisées : la DCT et la DWT. Les propriétés des coefficients de ces transformations sont analysées puis les bases théoriques sont introduites pour générer des vecteurs caractéristiques à partir de ces coefficients.

L'histogramme des vecteurs caractéristique est choisi comme le descripteur d'images. Ainsi, les concepts d'histogramme sont présentés.

En outre, l'algorithm K-means et les représentations parcimonieuses utilisés pour générer des histogrammes sont introduit. L'histogramme basé sur la représentation parcimonieuse est utilisé. Étant différent de l'histogramme classique, cet histogramme fournit plus d'informations sur la relation entre un vecteur et les vecteurs de base connexes.

Après la construction des descripteurs d'images, la mesure de similarité doit être considérée ; ainsi, nous avons présenté les mesures de similarité communément utilisées pour le CBIR, à partir desquelles deux types de distances, la distance de Manhattan et la distance de χ^2 sont choisies pour nos approches car correspondant à une faible charge de calcul.

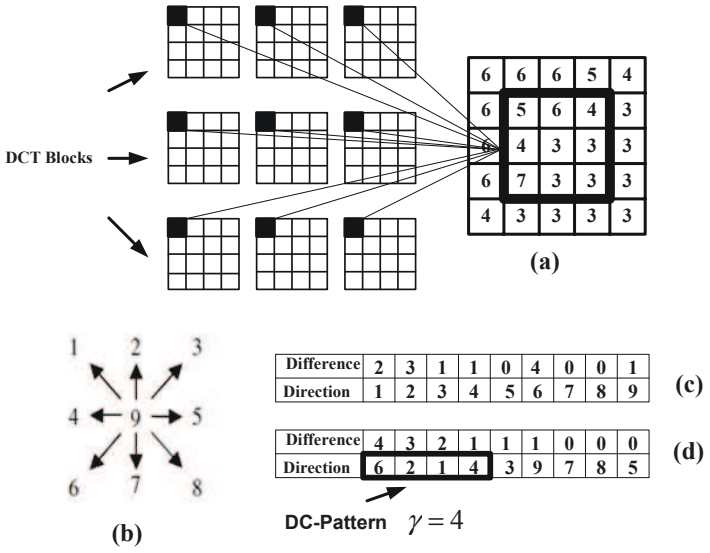
Enfin, l'évaluation de performance en CBIR, en particulier pour la reconnaissance de visage et la recherche de texture, est présentée, en s'appuyant sur la précision et le rappel avec ARR et EER car largement utilisés dans ce domaine.

Chapitre 3 : Descripteurs d'images dans le domaine DCT

Dans ce chapitre, après avoir présenté les travaux connexes, une approche améliorée et deux nouvelles approches sont proposées. Nous nommons ces trois approches respectivement : *Zigzag-Pattern*, *Sum-Pattern* et *Texture-Pattern*.

Les images sont d'abord décomposées en bloc DCT 4×4. Du fait que la même scène prise à différent niveaux de luminance conduira à différents blocs DCT, des étapes de prétraitement pour normaliser la luminance sont donc effectuées avant l'extraction de vecteurs caractéristiques. Elles sont réalisées par une mise à l'échelle des coefficients DCT en fonction du niveau de luminance moyen des coefficients DC des blocs DCT.

DC-Pattern est défini pour un ensemble de directions ayant les plus grandes différences entre la valeur du bloc courant et les valeurs des DC des blocs voisins. Huit différences entre le coefficient DC du bloc courant et celui de ses 8 voisins sont calculées. La neuvième est la différence entre la valeur DC du bloc courant et la moyenne de toutes les valeurs DC des neuf voisins (lui-même inclus). Les valeurs absolues de ces différences sont rangées par ordre décroissant et les γ premières directions avec les plus grandes différences forment le DC-Pattern. Ici γ est un paramètre qui peut être réglé pour obtenir un meilleur résultat de recherche. Le processus de construction du DC-Pattern est montré dans la Figure A.3.



(a) DC coefficients are extracted from neighboring blocks (b) Definition of directions
(c) Differences for each direction (d) Directions with largest differences form DC-Patterns

FIGURE A.3 – DC-Pattern construction

Un bloc DCT sans coefficient DC définit un AC-Pattern. Il existe deux méthodes

de balayage pour organiser les coefficients de l'AC-Pattern. La première façon consiste à balayer ligne par ligne, elle est aussi appelée balayage linéaire. Les coefficients AC sont classés de gauche à droite et de haut en bas. La seconde méthode est le balayage en zigzag que nous proposons d'utiliser. Ces deux méthodes sont représentées sur la Figure A.4.

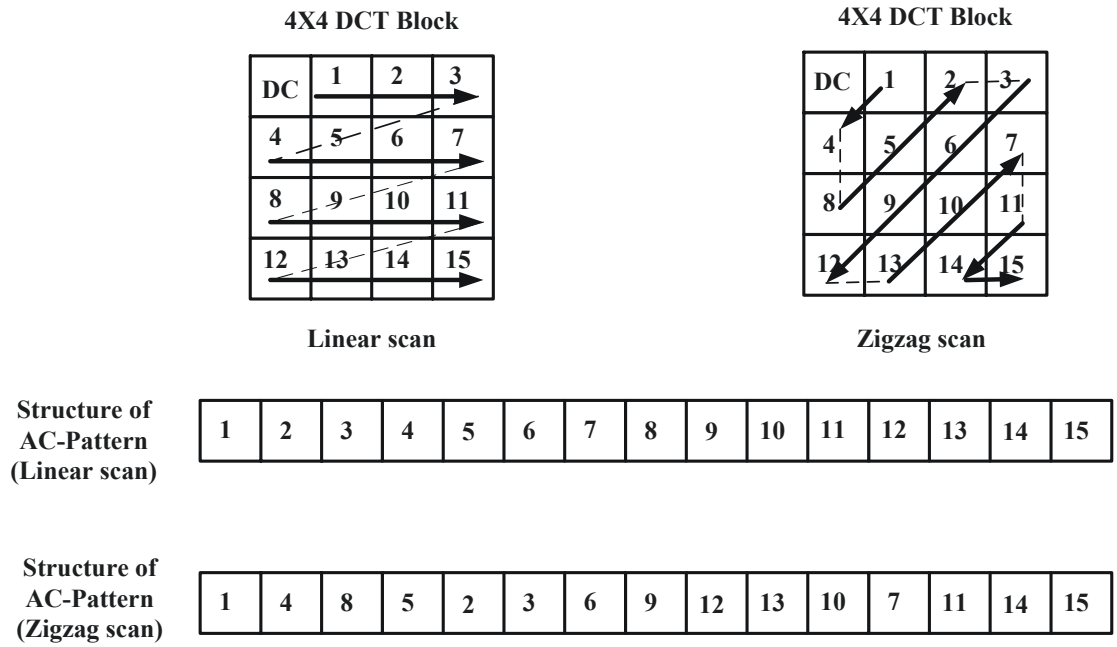


FIGURE A.4 – Balayage linéaire et zigzag

Comme les AC-Patterns des blocs observés sont nombreux, nous souhaitons en réduire le nombre. Aussi des motifs adjacents sont définis et fusionnés. Des AC-Patterns i et j sont dits adjacents si :

$$|C_i(1) - C_j(1)| \leq Th \textbf{ or } |C_i(2) - C_j(2)| \leq Th \textbf{ or } \cdots \textbf{ or } |C_i(m) - C_j(m)| \leq Th \quad \text{(A.16)}$$

où $C_i(k)$ $C_j(k)$ $(1 \leq k \leq m$, m indique le nombre de coefficients AC-Pattern) représentent les coefficients AC dans AC-Pattern. Th est le seuil. Dans notre méthode, $Th = 1$. Nous avons nommé *Zigzag-Pattern* cette approche de la construction d'AC-Patterns et de la production d'histogramme.

Dans le chapitre 1, il est rappelé que le coefficient DC indique l'énergie moyenne du bloc et que certains coefficients AC contiennent les informations directionnelles. Inspiré de cela, nous proposons deux types de vecteurs caractéristiques à partir de coefficients AC : *Sum-Pattern* et *Texture-Pattern*.

Nous sélectionnons 9 coefficients AC dans chaque bloc pour construire le Sum-Pattern. Ces 9 coefficients sont classés en 3 groupes : horizontal, vertical et diagonal. Les sommes de 2 ou 3 coefficients dans chaque groupe forment le Sum-Pattern. Le procédé de construction de cette structure est représenté sur la Figure A.5.

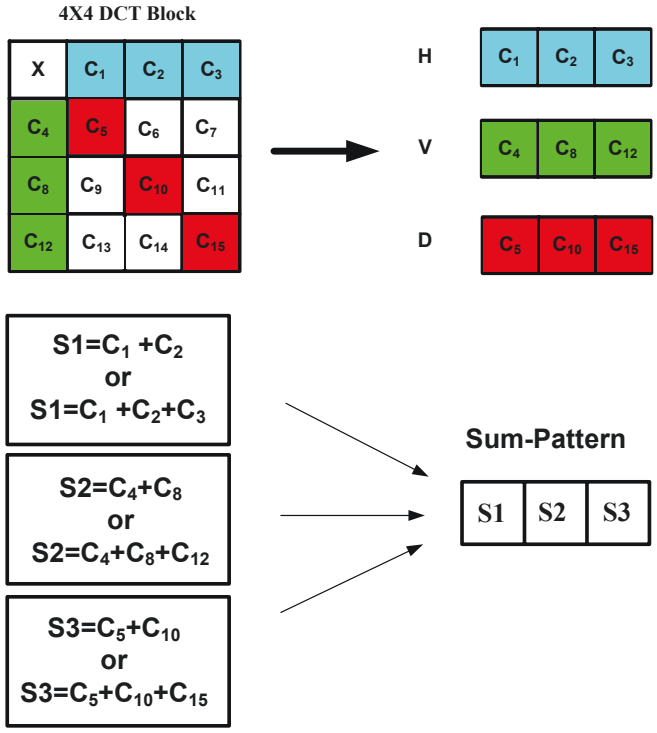


FIGURE A.5 – Sum-Pattern construction

Texture-Pattern est basé sur Sum-Pattern. 9 coefficients sont également classés en 3 groupes. Pour chaque groupe, la somme des coefficients est tout d'abord calculée et ensuite les différences au carré entre chaque coefficient et la somme de ce groupe sont calculées. Enfin, les sommes de ces différences au carré de chaque groupe sont utilisées pour construire le Texture-Pattern. Cette structure est représentée sur la Figure A.6.

Pour la recherche de texture couleur, Color-Pattern est construit par les coefficients DC des 3 composantes de chaque bloc dans l'image couleur en supposant qu'ils sont dans l'espace couleur YCbCr. La construction du Color-Pattern est illustrée à la Figure A.7.

L'histogramme de ces vecteurs de caractéristiques proposés est choisi comme le descripteur des images pour la reconnaissance faciale et de la recherche de la texture. La distance de Manhattan et la distance χ^2 sont choisies pour la mesure de similarité.

Ces approches sont appliquées dans la reconnaissance des visages et la recherche de texture (en version monochrome ou en version couleur) et conduisent à trois contributions :

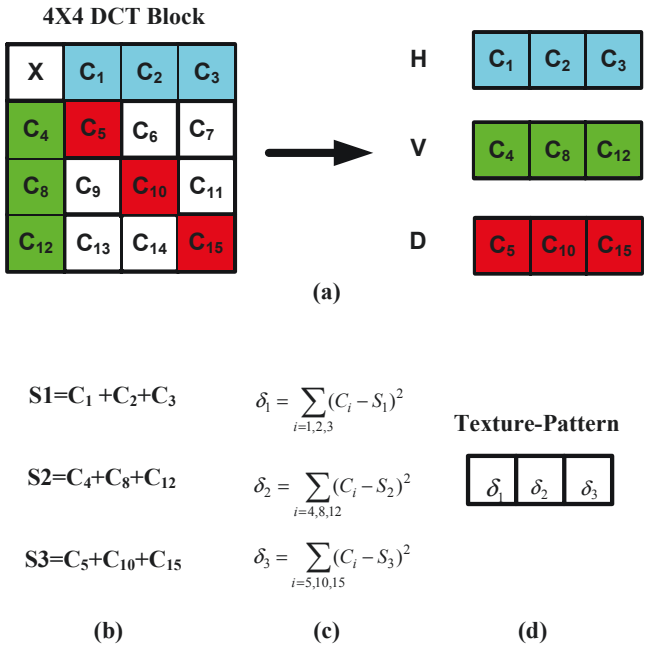


FIGURE A.6 – Texture-Pattern :
(a) Trois groupes de coefficients AC extraits d'un bloc DCT (b) Sommes de chaque groupe (c) Sommes de différences au carré (d) Texture-Pattern

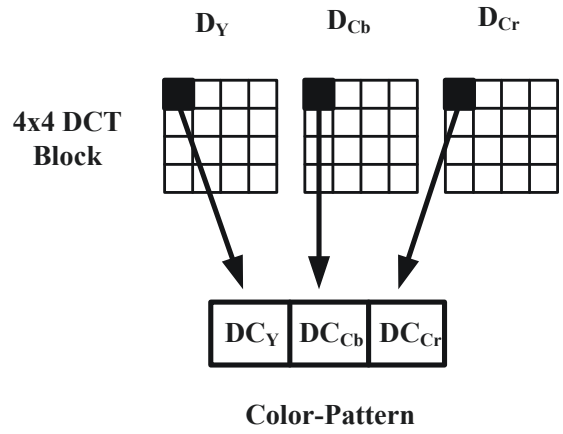


FIGURE A.7 – Color-Pattern

1. Les quatre nouveaux types de vecteurs de caractéristiques proposés : *Zigzag-Pattern*, *Sum-Pattern*, *Texture-Pattern* et *Color-Pattern*. Grâce à la capacité de DCT pour le compactage de l'énergie, et comme certains coefficients AC représentent la structure directionnelle de l'image, seulement quelques coefficients sont suffisants pour la construction de la caractéristique.

2. La fusion des motifs adjacents avec une sélection des motifs les plus fréquents pour

réduire la dimension des descripteurs de caractéristiques.

3. L'application de ces nouveaux types de vecteurs de caractéristiques a deux problématiques : la reconnaissance de visage avec les contenus structurels ; et la recherche de texture, avec les contenus structurels et non structurées.

Les résultats expérimentaux obtenus sur les trois bases de données du visage (ORL, GTF et FERET) et deux bases de données de texture (Small VisTex et VisTex entier) confirment l'efficacité de ces propositions.

Chapitre 4 : Descripteurs d'images dans le domaine des ondelettes

La transformée en ondelettes discrète est un outil pour extraire les caractéristiques des images. Dans ce chapitre, deux approches pour la recherche d'images de texture couleur dans le domaine des ondelettes sont proposés. Les ondelettes ont été largement appliquées pour la recherche de texture, ainsi un bref résumé des travaux connexes est d'abord présenté avant ces deux approches.

Dans nos deux méthodes, les images en couleur sont d'abord converties dans l'espace couleur YCbCr, dont les composantes sont I_Y, I_{Cb} et I_{Cr}. Puis, chaque composant est décomposée par DWT à N-niveau. Les résultats sont représentés par W_S^{mn}, où $S \in \{Y, Cb, Cr\}$ désigne les composantes et $m \in \{LL, HL, LH, HH\}$ les orientations de sous-bande, avec $n = \{1, 2, \ldots, N\}$ le niveau de décomposition. Dans notre approche, nous choisissons les ondelettes CDF 9/7.

4.1 Descripteur de texture couleur généré par K-means

Notre approche va s'appuyer sur une caractéristique intégrant texture et couleur qui va inclure deux vecteurs : vecteur texture et vecteur couleur. Les vecteurs texture multirésolution comprennent des coefficients ayant la même position spatiale dans les sous-bandes de détail de chaque niveau de décomposition sur le composante Y. D'autre part, le vecteur couleur est construit à partir des coefficients à la même position dans les sous-bandes d'approximation des composantes Y, Cb et Cr. La Figure A.8 montre le processus (N = 2, par exemple) et la définition du vecteur texture et du vecteur couleur qui sont répertoriés comme suit :

$$\begin{aligned} CV &= [W_Y^{LLN}(x_C, y_C),\ W_{Cb}^{LLN}(x_C, y_C),\ W_{Cr}^{LLN}(x_C, y_C)] \\ TV_n &= [W_Y^{HLn}(x_n, y_n),\ W_Y^{LHn}(x_n, y_n),\ W_Y^{HHn}(x_n, y_n)] \end{aligned} \tag{A.17}$$

où (x_n, y_n), $n = \{1, 2, \ldots, N\}$ et (x_C, y_C) indiquent les coordonnées du coefficient dans chaque sous-bande.

L'histogramme de ces vecteurs est choisi comme le descripteur de l'image. L'algorithme K-means est utilisé pour partitionner l'espace du vecteur couleur et du vecteur texture en partitions représentées par les centres de classe générés par K-means.

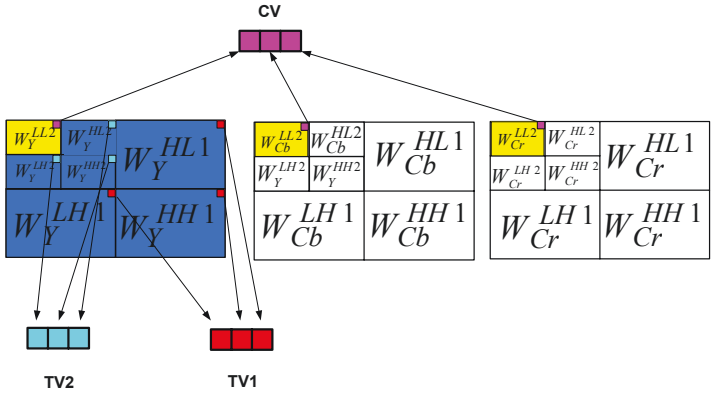


FIGURE A.8 – Projection des coefficients en vecteurs (N=2)

4.2 Descripteur de textures couleur généré par la représentation parcimonieuse

Deux problèmes principaux doivent être pris en compte pour l'application de l'histogramme basé sur la représentation parcimonieuse : le vecteur des caractéristiques et le dictionnaire, c'est pourquoi des vecteurs des caractéristiques multirésolution et un dictionnaire en ondelettes sont proposés.

Les vecteurs de caractéristiques construits à partir des coefficients de chaque sous-bande sont classés en deux types : un vecteur d'approximation A et un vecteur de détail T. A est construit à partir de W_S^{LLN} dont les coefficients sont à la même position pour chaque composante de couleur. T est construit à partir de trois sous-bandes de détails à chaque niveau de décomposition et dont les éléments sont les coefficients au même emplacement dans chacune des sous-bandes horizontales, verticales et diagonales et cela au même niveau de décomposition. La Figure A.9 illustre cette procédure lorsque N = 2, les définitions des vecteurs caractéristiques sont les suivantes :

$$\begin{aligned} A = CV = [&W_Y^{LLN}(x_a, y_a),\ W_{Cb}^{LLN}(x_a, y_a),\ W_{Cr}^{LLN}(x_a, y_a)] \\ Tn = [&W_Y^{HLn}(x_n, y_n),\ W_Y^{LHn}(x_n, y_n),\ W_Y^{HHn}(x_n, y_n), \\ &W_{Cb}^{HLn}(x_n, y_n),\ W_{Cb}^{LHn}(x_n, y_n),\ W_{Cb}^{HHn}(x_n, y_n), \\ &W_{Cr}^{HLn}(x_n, y_n),\ W_{Cr}^{LHn}(x_n, y_n),\ W_{Cr}^{HHn}(x_n, y_n)] \end{aligned} \tag{A.18}$$

L'algorithme K-means est appliqué pour générer le dictionnaire à partir de l'ensemble

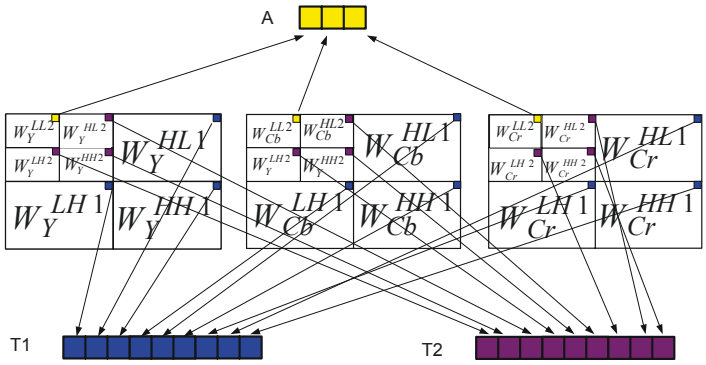


FIGURE A.9 – Projection des coefficients en vecteurs multiresolution (N=2)

d'apprentissage : l'ensemble des centres des partitionnement des vecteurs d'apprentissage résultant des K-means sont utilisés comme dictionnaire.

4.3 Mesure de similarité et évaluation des performances

Nous avons choisi la distance de χ^2 pour mesurer la similarité entre l'histogramme de la requête et l'histogramme de une image de la base de données. Pour évaluer la performance de chaque approach, des expériences sont mises en oeuvre sur quatre bases de données : Small VisTex, VisTex ensemble, ALOT et STex. Comparée avec l'état de l'art, la méthode basée sur K-means n'obtient pas toujours les meilleures performances tandis que la méthode basée sur la représentation parcimonieuse surpasse toujours. Mais cependant la méthode basée sur K-means ne reste pas un mauvais choix pour la recherche de texture couleur si l'on tient compte du nombre réduit de coefficients d'ondelettes utilisé.

4.4 Conclusion

Deux approaches pour la recherche de texture couleur dans le domaine d'ondelettes sont donc présentés. Ces propositions conduisent à quatre contributions :

1. La proposition de vecteurs texture multirésolution et de vecteur couleur traitant seulement 50% au maximum des coefficients pour la construction de vecteurs de caractéristiques.

2. La dimension des descripteurs de caractéristiques est réduite avec l'utilisation de l'algorithme K-means pour diviser l'espace des vecteurs des caractéristiques en partitions.

3. Une nouvelle approche pour la description de texture couleur basée sur la représentation parcimonieuse est proposée. Ce descripteur est plus précis, conduisant à de meilleures performances dans la recherche de texture couleur.
4. Cette approche de description de textures couleur est beaucoup moins coûteuse que celle de la recherche probabiliste de texture couleur concernant les aspects de calcul de la similarité entre les descripteurs.

Chapitre 5 : Conclusion et perspectives

Dans cette thèse, nous avons introduit, analysé et étudié l'analyse d'images pour une recherche d'images basée contenu dans le domaine transformé. Ce travail se concentre principalement sur les algorithmes basés DCT et DWT, largement utilisées dans la compression des images et des vidéos.

Le contexte et quelques concepts fondamentaux liés au CBIR et à nos propositions ont tout d'abord été introduit. En général, deux types de fonctions sont utilisés pour la recherche d'images : l'un basé sur l'intensité (couleur et texture) et l'autre sur la géométrie (forme). Nous nous sommes concentrés sur l'intensité.

Ensuite, nous avons proposé une approche améliorée d'une approche existante et deux nouvelles approches dans le domaine DCT. Cette approche améliorée est basée sur une méthode dans laquelle l'histogramme de l'AC-Pattern et celui du DC-Pattern sont utilisés comme descripteur de caractéristique. Deux améliorations ont été apportées : le balayage en zigzag est utilisé pour organiser les coefficients d'AC-Pattern et des motifs adjacents sont définis et fusionnés. Compte-tenu des caractéristiques des coefficients DCT, deux types de vecteurs caractéristiques sont alors proposés : *Sum-Pattern* and *Texture-Pattern*. Les deux vecteurs des caractéristiques sont des vecteurs 3-D qui représentent les informations directionnelles du bloc DCT et peuvent être appliqués à la fois dans la reconnaissance des visages et la recherche de textures. Enfin, *Color-Pattern* est proposé et utilisé en conjonction avec *Texture-Pattern* pour la recherche de textures couleur. Les résultats expérimentaux obtenus des échantillons communément adoptés dans les bases d'images de visages, ORL, GTF, FERET, ainsi que la base de données de texture standard, VisTex, ont démontré la l'apport de nos approches.

Enfin, nous avons proposé deux approches pour la recherche de texture couleur dans le domaine des ondelettes. La première est dans le contexte de la construction séparée de caractéristiques de couleur et de texture : les caractéristiques de couleur sont extraites par les coefficients de la sous-bande d'approximation de DWT, celles des textures sont construites avec les coefficients des sous-bandes DWT dans la composante de luminance des textures. L'avantage principal de cette proposition est qu'elle ne nécessite le traitement que de 50% des coefficients d'ondelettes. La seconde approche se situe dans le contexte de l'utilisation conjointe de la couleur et de la texture. Les avantages ont deux aspects : le premier est qu'un vecteur de caractéristiques sera représenté par un certain nombre

de vecteurs de base dans l'histogramme permettant ainsi une représentation plus précise qu'un unique vecteur ; le second est que l'observation de la similarité entre les descripteurs est beaucoup moins coûteuse en calculs. Les expériences ont été exécutées sur quatre bases des données de texture couleur : Small VisTex, VisTex ensemble, ALOT et STex. Tous ces résultats ont confirmé les contributions de ces deux propositions.

Les perspectives de travaux futurs pourront se concentres sur des aspects suivants :

1. Extension de l'histogramme basé sur la représentation parcimonieuse au domaine DCT.
2. Vérification de la capacité des vecteurs des caractéristiques multirésolution dans le domaine DWT pour l'extraction des images multiresolution.
3. Recherche d'un dictionnaire plus compact.
4. Intégration de plus de caractéristiques dans le cadre de l'histogramme basé sur représentation parcimonieuse.

List of Figures

List of Tables

List of papers

Journals:

[1] **Cong Bai**, Wenbin Zou, Kidiyo Kpalma, Joseph Ronsin, "Efficient color texture image retrieval by combination of color and texture features in wavelet domain," *Electronics Letters*, 48, 1463-1465, 2012

http://hal.archives-ouvertes.fr/hal-00776192/

[2] **Cong Bai**, Weizhi Lu, Kidiyo Kpalma, Joseph Ronsin, "Color texture retrieval using sparse representation based histogram," (Submitted for *IEEE transaction on Image Processing*)

Papers in books as a chapter:

[3] **Cong Bai**, Kidiyo Kpalma, Joseph Ronsin, "Analysis of histogram descriptor for image retrieval in DCT domain," *Intelligent Interactive Multimedia Systems and Services*, 2011, Springer, ISBN: 978-3-642-22157-6

http://hal.archives-ouvertes.fr/hal-00611840

[4] Kidiyo Kpalma, **Cong Bai**, Miloud Chikr El-Mezouar, Kamel Belloulata, "A New Histogram-based Descriptor for Images Retrieval from databases," To appear on *Studies in Computational Intelligence*，Springer

http://hal.archives-ouvertes.fr/hal-00728214

Conference papers:

[5] **Cong Bai**, Kidiyo Kpalma, Joseph Ronsin, "A new descriptor based on 2D DCT for image retrieval," International Conference on Computer Vision Theory and Applications (Visapp2012), 714-717, Feb. 2012

http://hal.archives-ouvertes.fr/hal-00728076

[6] **Cong Bai**, Kidiyo Kpalma, Joseph Ronsin, "Color textured image retrieval by combining texture and color features," Proceedings of the 20th European Signal Processing Conference (EUSIPCO2012), pp.170-174, Aug. 2012

http://hal.archives-ouvertes.fr/hal-00728083

[7] **Cong Bai**, Kidiyo Kpalma, Joseph Ronsin, "An improved feature vector for content-based image retrieval in DCT domain," International Conference on Computer Vision Theory and Applications (Visapp2013), Feb. 2013

http://hal.archives-ouvertes.fr/hal-00799244

Bibliography

[1] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 1, pp. 39 – 62, 1999.

[2] H. Tamura and N. Yokoya, "Image database systems: A survey," *Pattern Recognition*, vol. 17, no. 1, pp. 29 – 43, 1984. Knowledge Based Image Analysis.

[3] S.-K. Chang and A. Hsu, "Image information systems: where do we go from here?," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 4, pp. 431 –442, oct 1992.

[4] J. Z. Wang, N. Boujemaa, A. Del Bimbo, D. Geman, A. G. Hauptmann, and J. Tesić, "Diversity in multimedia information retrieval research," in *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, MIR '06, (New York, NY, USA), pp. 5–12, ACM, 2006.

[5] J. Smith and S.-F. Chang, "Visually searching the web for content," *MultiMedia, IEEE*, vol. 4, pp. 12 –20, jul-sep 1997.

[6] S. Sclaroff, L. Taycher, and M. LaCascia, "Imagerover: A content-based image browser for the world wide web," tech. rep., Boston University, Boston, MA, USA, 1997.

[7] Google, "Google image search." http://images.google.com/.

[8] I. Ahmad, S. Abdullah, S. Kiranyaz, and M. Gabbouj, "Content-based image retrieval on mobile devices," in *Proc. SPIE 5684, Multimedia on Mobile Devices*, pp. 255–264, 2005.

[9] M. La Cascia, M. Morana, and S. Sorce, "Mobile interface for content-based image management," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, pp. 718 –723, feb. 2010.

[10] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang, "Supporting similarity queries in mars," in *Proceedings of the fifth ACM international conference on Multimedia*, MULTIMEDIA '97, (New York, NY, USA), pp. 403–413, ACM, 1997.

[11] A. Lumini and D. Maio, "Haruspex: an image database system for query-by-examples," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, pp. 258 –261 vol.4, 2000.

[12] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applications -clinical benefits and future directions," *International Journal of Medical Informatics*, vol. 73, no. 1, pp. 1 – 23, 2004.

[13] H.-c. Cho, L. Hadjiiski, B. Sahiner, H.-P. Chan, C. Paramagul, M. Helvie, and A. V. Nees, "Interactive content-based image retrieval (CBIR) computer-aided diagnosis (cadx) system for ultrasound breast masses using relevance feedback," in *Proc. SPIE 8315, Medical Imaging 2012: Computer-Aided Diagnosis*, pp. 831509–831509–7, 2012.

[14] P. M. Kelly, T. M. Cannon, and D. R. Hush, "Query by image example: the comparison algorithm for navigating digital image databases (candid) approach," in *Proc. SPIE 2420, Storage and Retrieval for Image and Video Databases III*, pp. 238–248, 1995.

[15] S. Aksoy and R. Haralick, "Probabilistic vs. geometric similarity measures for image retrieval," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 357 –362 vol.2, 2000.

[16] H.-L. Peng and S.-Y. Chen, "Trademark shape recognition using closed contours," *Pattern Recognition Letters*, vol. 18, no. 8, pp. 791 – 803, 1997.

[17] P.-Y. Yin and C.-C. Yeh, "Content-based retrieval from trademark databases," *Pattern Recognition Letters*, vol. 23, no. 13, pp. 113–126, 2002.

[18] J. Schietse, J. P. Eakins, and R. C. Veltkamp, "Practice and challenges in trademark image retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, (New York, NY, USA), pp. 518–524, ACM, 2007.

[19] H. Qi, K. Li, Y. Shen, and W. Qu, "An effective solution for trademark image retrieval by combining shape description and feature matching," *Pattern Recognition*, vol. 43, no. 6, pp. 2017 – 2027, 2010.

[20] W. Wang, C.-H. Wei, L. Zhang, and X. Wang, "Traffic-signs recognition system based on multi-features," in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2012 IEEE International Conference on*, pp. 120 –123, july 2012.

[21] Z. Geradts, *Content-based information retrieval from forensic image databases*. PhD thesis, Utrecht University, The Netherlands, June 2002.

[22] Y.-B. Lee, U. Park, A. K. Jain, and S.-W. Lee, "Pill-ID: Matching and retrieval of drug pill images," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 904 – 910, 2012.

[23] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1349 –1380, dec 2000.

[24] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, pp. 1–19, Feb. 2006.

[25] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262 – 282, 2007.

[26] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, pp. 5:1–5:60, May 2008.

[27] G. Rafiee, S. Dlay, and W. Woo, "A review of content-based image retrieval," in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pp. 775 –779, july 2010.

[28] M. Yang, *Shaped-based feature extraction and similarity matching*. These, INSA de Rennes, July 2008.

[29] T. Mäenpää and M. Pietikäinen, "Classification with color and texture: jointly or separately?," *Pattern Recognition*, vol. 37, no. 8, pp. 1629 – 1640, 2004.

[30] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," *Inf. Retr.*, vol. 11, pp. 77–107, Apr. 2008.

[31] O. A. Penatti, E. Valle, and R. da S. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 359 – 380, 2012.

[32] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. Norwell, MA, USA: Kluwer Academic Publishers, 1st ed., 1992.

[33] G. Strang, "The discrete cosine transform," *SIAM Rev.*, vol. 41, pp. 135–147, Mar. 1999.

[34] B. Shen, "From 8-tap dct to 4-tap integer-transform for mpeg to h.264/avc transcoding," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 1, pp. 115 – 118 Vol. 1, oct. 2004.

[35] I. Daubechies, *Ten lectures on wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[36] S. Mallat, "Applied mathematics meets signal processing," in *Proceedings of the International Congress of Mathematicians*, pp. 319–338, 1998.

[37] F. C. Tony and S. Jianhong, *Image Processing and Analysis*. Society for Industrial and Applied Mathematics, 2005.

[38] S. W. James, *A Primer on Wavelets and their scientific Appliactions Second edition*. Chapman Hall, 2007.

[39] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485–560, 1992.

[40] M. Unser and T. Blu, "Mathematical properties of the jpeg2000 wavelet filters," *Image Processing, IEEE Transactions on*, vol. 12, pp. 1080 – 1090, sept. 2003.

[41] S. Siggelkow, *Feature histograms for content based image retrieval*. PhD thesis, University Freiburg, 2002.

[42] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.

[43] P. Andritsos, "Data clustering techniques," *Toronto, University of Toronto, Dep. of Computer Science*, vol. 1, no. 1, pp. 3–2, 2002.

[44] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Mach. Learn.*, vol. 56, pp. 9–33, June 2004.

[45] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, pp. 159–179, 1985.

[46] C. A. Sugar, L. A. Lenert, and R. A. Olshen, "An application of cluster analysis to health services research: Empirically defined health states for depression from the sf-12," tech. rep., Stanford University, 1999.

[47] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[48] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, pp. 1031 –1044, june 2010.

[49] D.Lee and H.Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788 –791, oct. 1999.

[50] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[51] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19 – 60, 2010.

[52] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[53] Julien Mairal and Francis Bach and Jean Ponce, "SPArse Modeling Software." `http://spams-devel.gforge.inria.fr/index.html`. Online; accessed June 2011.

[54] Y. Rubner, *Perceptual Metrics For Image Database Navigation*. PhD thesis, Stanford University, Stanford, CA, USA, 1999.

[55] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11–32, 1991.

[56] S. Kullback, *Information theory and statistics*. Dover Publications, New York, 1968.

[57] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, pp. 231–262, 1994.

[58] Y. Rubner and C. Tomasi, *Perceptual Metrics for Image Database Navigation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[59] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, pp. 99–121, Nov. 2000.

[60] R. Picard, T. Kabir, and F. Liu, "Real-time recognition with the entire brodatz texture database," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pp. 638 –639, jun 1993.

[61] S. Krishnamachari and M. Abdel-Mottaleb, "Color compact descriptor for fast image and video segment retrieval," in *Storage and Retrieval for Media Databases'00*, pp. 581–589, 2000.

[62] R. Bolle, S. Pankanti, and N. Ratha, "Evaluation techniques for biometrics-based authentication systems (frr)," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2, pp. 831 –837 vol.2, 2000.

[63] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *Int. J. Comput. Vision*, vol. 43, pp. 167–188, July 2001.

[64] D. Ramasubramanian and Y. Venkatesh, "Encoding and recognition of faces based on the human visual model and dct," *Pattern Recognition*, vol. 34, no. 12, pp. 2447 – 2458, 2001.

[65] S. Dabbaghchian, M. P. Ghaemmaghami, and A. Aghagolzadeh, "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology," *Pattern Recognition*, vol. 43, no. 4, pp. 1431 – 1440, 2010.

[66] M. Shneier and M. Abdel-Mottaleb, "Exploiting the jpeg compression scheme for image retrieval," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 849 –853, aug 1996.

[67] A. Nefian and I. Hayes, M.H., "Hidden markov models for face recognition," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 5, pp. 2721 –2724 vol.5, may 1998.

[68] S. Eickeler, S. Müller, and G. Rigoll, "Recognition of jpeg compressed face images based on statistical methods," *Image and Vision Computing*, vol. 18, no. 4, pp. 279 – 287, 2000.

[69] D. Zhong and I. Defee, "Pattern recognition in compressed dct domain," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 3, pp. 2031 – 2034 Vol. 3, oct. 2004.

[70] D. Zhong and I. Defée, "DCT histogram optimization for image database retrieval," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2272 – 2281, 2005.

[71] G. Feng and J. Jiang, "JPEG compressed image retrieval via statistical features," *Pattern Recognition*, vol. 36, pp. 977–985, 2003.

[72] T. Tsai, Y.-P. Huang, and T.-W. Chiang, "Image retrieval based on dominant texture features," in *Industrial Electronics, 2006 IEEE International Symposium on*, vol. 1, pp. 441 –446, july 2006.

[73] P. Poursistani, H. Nezamabadi-pour, R. A. Moghadam, and M. Saeed, "Image indexing and retrieval in JPEG compressed domain based on vector quantization," *Mathematical and Computer Modelling*, no. 0, pp. –, 2011.

[74] A. Vellaikal and C.-C. Kuo, "Joint spatial-spectral indexing for image retrieval," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, pp. 867 –870 vol.3, sep 1996.

[75] C.-W. Ngo, T.-C. Pong, and R. T. Chin, "Exploiting image indexing techniques in DCT domain," *Pattern Recognition*, vol. 34, no. 9, pp. 1841 – 1851, 2001.

[76] C. Theoharatos, V. Pothos, G. Economou, and S. Fotopoulos, "Compressed domain image indexing and retrieval based on the minimal spanning tree," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pp. 1516 –1519, july 2005.

[77] C. Theoharatos, V. Pothos, N. Laskaris, G. Economou, and S. Fotopoulos, "Multivariate image similarity in the compressed domain using statistical graph matching," *Pattern Recognition*, vol. 39, no. 10, pp. 1892 – 1904, 2006.

[78] Z. ming Lu and H. Burkhardt, "A content-based image retrieval scheme in jpeg compressed domain," *International Journal of Innovative Computing, Information and Control*, vol. 2, pp. 831–839, August 2006.

[79] D. Zhong, "Image database retrieval methods based on feature histograms," *Tampereen teknillinen yliopisto. Julkaisu-Tampere University of Technology. Publication; 734*, 2008.

[80] Georgia Tech, "GTF database." `http://www.anefian.com/research/face_reco.htm`. Online; accessed March 2010.

[81] AT and T Laboratories Cambridge, "ORL database." http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. Online; accessed March 2010.

[82] M. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance," *Image Processing, IEEE Transactions on*, vol. 11, pp. 146 –158, feb 2002.

[83] M. Kokare, P. Biswas, and B. Chatterji, "Texture image retrieval using new rotated complex wavelet filters," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, pp. 1168 –1178, dec. 2005.

[84] R. Kwitt and A. Uhl, "Lightweight probabilistic texture retrieval," *Image Processing, IEEE Transactions on*, vol. 19, pp. 241 –253, jan. 2010.

[85] M. Media Laboratory, "Vistex database of textures." http://vismod.media.mit.edu/vismod/imagery/VisionTexture/. Online; accessed Dec. 2010.

[86] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1090 – 1104, oct 2000.

[87] Y. Stitou, N. Lasmar, and Y. Berthoumieu, "Copulas based multivariate gamma modeling for texture classification," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1045 –1048, april 2009.

[88] L. Bombrun, Y. Berthoumieu, N.-E. Lasmar, and G. Verdoolaege, "Multivariate texture retrieval using the geodesic distance between elliptically distributed random variables," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 3637 –3640, sept. 2011.

[89] A. Jain and G. Healey, "A multiscale representation including opponent color features for texture recognition," *Image Processing, IEEE Transactions on*, vol. 7, pp. 124 –128, jan 1998.

[90] C. L. Xu and X. T. Zhen, "Chromatic statistical landscape features for retrieval of color textured images," in *Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on*, pp. 98 –101, dec. 2009.

[91] N. Vasconcelos and A. Lippman, "Library-based coding: a representation for efficient video compression and retrieval," in *Data Compression Conference, 1997. DCC '97. Proceedings*, pp. 121 –130, mar 1997.

[92] N. Vasconcelos and A. Lippman, "A unifying view of image similarity," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, pp. 38 –41 vol.1, 2000.

[93] N. Vasconcelos and A. Lippman, "A probabilistic architecture for content-based image retrieval," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1, pp. 216 –221 vol.1, 2000.

[94] G. Verdoolaege, S. De Backer, and P. Scheunders, "Multiscale colour texture retrieval using the geodesic distance between multivariate generalized gaussian models," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 169 –172, oct. 2008.

[95] R. Kwitt, P. Meerwald, and A. Uhl, "Efficient texture image retrieval using copulas in a bayesian framework," *Image Processing, IEEE Transactions on*, vol. 20, pp. 2063 –2077, july 2011.

[96] T. Chen, K.-K. Ma, and L.-H. Chen, "Discrete wavelet frame representations of color texture features for image query," in *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pp. 45 –50, dec 1998.

[97] T. Yumin and M. Lixia, "Image retrieval based on multiple features using wavelet," in *Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on*, pp. 137 – 142, sept. 2003.

[98] S. Liapis and G. Tziritas, "Color and texture image retrieval using chromaticity histograms and wavelet frames," *Multimedia, IEEE Transactions on*, vol. 6, pp. 676 – 686, oct. 2004.

[99] Y. D. Chun, N. C. Kim, and I. H. Jang, "Content-based image retrieval using multiresolution color and texture features," *Multimedia, IEEE Transactions on*, vol. 10, pp. 1073 –1084, oct. 2008.

[100] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." `http://www.vlfeat.org/`, 2008. Online; accessed May 2012.

[101] G. J. Burghouts and J.-M. Geusebroek, "Material-specific adaptation of color invariant features," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 306 – 313, 2009.

[102] W. of University of Salzburg, "Salzburg texture image database." http://www.wavelab.at/sources/STex/. Online; accessed Sep. 2012.

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Image analysis for content based image retrieval in transform domain

Nom Prénom de l'auteur : BAI CONG

Membres du jury :
- Monsieur CARRE Philippe
- Monsieur RONSIN JOSEPH
- Monsieur KPALMA Kidiyo
- Monsieur TALEB Nasreddine
- Monsieur POISSON Gérard

Président du jury : Philippe Carré

Date de la soutenance : 21 Février 2013

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 21 Février 2013

Signature du président de jury
P. Carré

Le Directeur,

M'hamed DRISSI